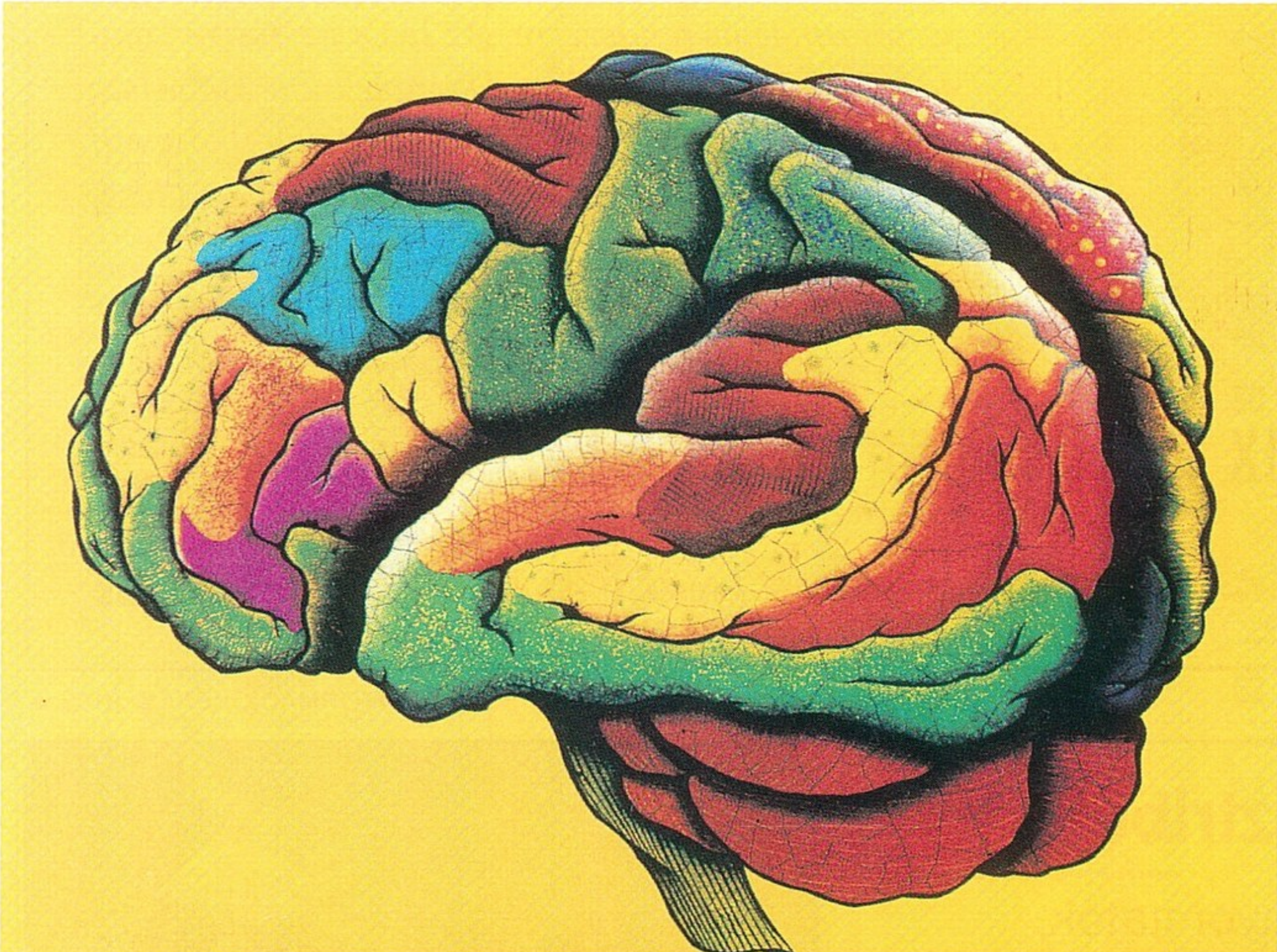
1996 / JANUÁR

ÁRA: 356 FT

ÚJ ALAPLAP

SZÁMÍTÁSTECHNIKAI FOLYÓIRAT LEMEZMELLÉKLETTEL





Mostantól
3,5" HD, 1,44 MB

A HÓNAP TÉMÁJA:

MEMÓRIAZAVAROK

Alternatív ablakok

1995. évi összesített tartalomjegyzék

Napirenden ismét az órarend

ÚJ ALAPLAP

A Mikroszámítógép Magazin és az Alaplap hagyományait folytató számítástechnikai folyóirat

Megjelenik havonta, mágneslemez melléklettel

Főszerkesztő:
Faklen Pál

Főszerkesztő-helyettes:
Varga János

Szerkesztő:
Jakab Ágnes

A szerkesztőbizottság tagjai:
Aszalós László, Csórián Sándor, Feleki Zoltán, Ferenczi Gábor, Herczeg József, Horlai János, Kis János, Nagy Gábor, Pogány Csaba, Szondi Egon János, Vargha Dénes, Vékony Tamás

Szerkesztőség és kiadó:
1539 Budapest I., Márvány u. 17.
Telefon: 156-3211 / 200, 214
Fax (manuális): 156-3211 / 201
E-mail: alaplap@mail.datanet.hu

Felelős kiadó:
Faklen Pál

Terjesztés:
Megyes Zsuzsanna

Hirdetésszervezés:
Árvai Katalin, Bogácsi Mária, Galyasi Hedvig, Pap Katalin

Külföldi hirdetések:
PubliCity
Reklám- és Médiaügynökség
1537 Budapest I., Márvány u. 17.
Telefon: 156-1182 Fax: 175-3539

Példányszámadatok hitelesítése:
Magyar Terjesztésellenőrző Szövetség
MATESZ
és Price Waterhouse

Nyomtatás:
Zalai Nyomda Rt, Zalaegerszeg
Felelős vezető:
Somogyi Tibor ügyvezető igazgató

Terjeszti:
A Magyar Posta Rt, a Nemzeti Hírlapkereskedelmi Rt, a Hírker Rt, a Kiadói Lapterjesztő Kft, számos számítástechnikai szaküzlet és más alternatív terjesztő

Előfizethető a kiadónál:
Új Alaplap Kiadói Kft,
1539 Budapest, Pf. 571
Bankszámlaszám:
OTP 11701004-20171649
Eladási ár: 356 Ft
Évi előfizetési díj: 3564 Ft

Külföldre terjeszti a Kultúra,
H-1389 Budapest, Pf. 149

HU ISSN 1217-7598

A HÓNAP TÉMÁJA: MEMÓRIAZAVAROK

(Összeállította: Varga János)

Flip-flop

Vannak dolgok, amelyek egyidősek a számítógép megjelenésével, naponta használjuk, definíciójukat is tudjuk, mégis valamiféle misztérium övezi őket. És senkinek sem akaródzik a homályt eloszlatni, vagy a „levegőben lógó” kérdésekre a választ megadni.

Tulajdonképpen egy számítástechnikai lap számára snassz dolognak tűnhet, hogy éppen a számítógép lényegével összefüggésben kell feltenni például olyan kérdést, hogy mi is tulajdonképpen a memória. Pedig — úgy érezzük — a válaszadást ilyen triviális szintről kell elkezdeni.

Unos-untalan halljuk szakértő dilettánsok szájából: tudod, ilyenkor a számítógép *beolvassa az adatot a memóriájába*, és onnan *történik az adat feldolgozása*. (Mit is csinál? És mi csinálja? És hogy olvassa be?)

Célunk ezúttal (is) valamiféle demisztifikáció. Szeretnénk, ha a memória szó hallatán a dolog a maga fizikai valójában, működési mechanizmusával, korlátaival és képességeivel egyetemben jelenne meg. Szeretnénk például, ha egyre kevesebben ülnének fel a programszerzők által rutinszerűen beültetetett „Too big to fit in memory” vagy „Not enough memory” üzenetek félrevezetésének. Szeretnénk, ha mind többben tudnák: nem mindig kell hanyatthomlok memóriáért rohanni a boltba, mert nem biztos, hogy az elvégzendő feladat ezt megköveteli, lehet, hogy csak a programozó mohósága vagy lustasága van mögötte.

Világosan kell azonban azt is látni: e témakörben is érvényes a „valamit valamiért” elve. A valóban memóriaigényes grafikus alkalmazásoknál és a nagy számítási igényű feladatoknál nem lehet megspórolni a többletmemóriát, mert egyébként nem várható el zökkenőmentes munkavégzés.

E megfontolásból — s nem félve attól, hogy ránk sütik a „dedós” megközelítés bélyegét — beleütöttük az orrunkat sok mindenbe, ami a memóriával kapcsolatos, az összeállítás szerkezeti felépítésekor pedig a „flip-flop” technikát választottuk: egyenként felváltva követik egymást a szoftvermegközelítésű írások és a dolog hardveroldalát boncolgatók.

A memóriaproblémák vizsgálatakor ugyanis más számítástechnikai témákkal összehasonlítva sokkal inkább kell egyszerre jelen lenniük a hardver- és a szoftvermegfontolásoknak.

Ha kicsi a RAM-od...

Ria, Ria, MemóRia!

Memória. A számítástechnika benzinje.
Két éve leégett egy memóriachipekhez speciális gyantát előállító gyár a Távol-Keleten. Akkor duplájára emelkedett a RAM-ok ára. De mintha ott (magasan) felejtették volna...
Olvasom, hogy a legnagyobb processzorgyártó cég, az Intel éves profitrátája eléri a 60%-ot.
Szerintem a memóriagyártóké is (Samsung, IBM, NEC, Toshiba stb.).

Memória. Mindig kevés. Mindig több és több kell a memóriafaló grafikus alkalmazásokhoz. Vajon mindig így volt? Hát nem egészen. Emlékszünk még Lord Sinclair ZX 81-ére? Abban 4 Kbájt RAM volt csupán, mégis a ROM-BASIC-kel csodákat műveltek a beavatottak. Aztán jött a ZX Spectrum (48 Kbájt RAM), később a Commodore 64 — a nevében a memóriakapacitása. (A Commodore 16 nem lett akkora siker, miként a Commodore 128 sem.) Jöttek az Amigák, Atarik és Apple-ek. Ez volt a 8 bites processzorok aranykora. Kiváló játékgépek voltak, sokan ezeken sajátították el a számítástechnika, sőt a programozás alaplépéseit. De 1982-ben megjelent az IBM PC, majd PC XT, amelyben már két floppyegység, sőt merevlemez is volt. Általában 256 Kbájt RAM-mal szállították, de a nagyobb alkalmazások (például dBASE III Plus) már 512, sőt 640 Kbájtot követelt meg.

Jöttek az AT-k: 286 (1 Mbájt RAM, NEAT-chipset, hogy a 640 K feletti RAM is kihasználható legyen). A komolyabb alkalmazások már nem fértek bele az Intel mikroprocesszorok szabta 64 Kbájtos szegmensekbe, bár a 64 K máig alapegység a PC-s világban. Létrehozták a LIM (Lotus, Intel, Microsoft) EMS (expanded memory segment) szabványt, amelynek az a lényege, hogy a 640 Kbájt mellé tett memóriabővítést egy, a 640 Kbájt és 1 Mbájt közötti 64 K-s ablakon keresztül lássa 64 Kbájtonként az erre felkészített program (az elsők között: Lotus 1-2-3, FrameWork, Symphony stb.).

A Microsoft ügyesen kihasználta az Intel 286/386 processzor adta plusz lehetőséget is, miszerint fizikailag az 1 Mbájt fölé is tudott egy 64 Kbájtos területet címezni. Ezt hívták HMA-nak (High Memory Area), ahová a DOS-ok 5.0-tól kezdődően fel tudták tölteni magukat, ami által több hely maradt a konvencionális memóriaterületen.

A PC-s világ fokozatos és töretlen fejlődése — mint látjuk — megfigyelhető a memóriafelhasználás és kihasználás mikéntjeként is. A 386-os processzorok már az extended, 1 Mbájt feletti memóriát is ki tudták használni. Így a programozók kezdtek védett (ún. protected) módú programokat írni, ami lehetővé tette a multitaszkos és egyszerre nagy mennyiségű adatokat gyorsan feldolgozó programok írását.

Ahogy gyorsultak a PC-k (gyorsabb processzorok, gyorsabb alaplapok, ehhez ugye fürgébb RAM-ok kellettek), úgy valósult meg a PC-sek álma is (mert az Apple Macintoshához eredendően grafikus operációs rendszer járt).

Igen ám, de amíg egy betű karakteres ábrázolásához elég volt egy bájt, addig grafikus ábrázolásban, ahol maga a szín is lényeges információ, már két, kb. 3 nagyságrenddel több memória kellett az egyértelmű kódoláshoz.

Egy karakteres képernyő 25 sorból állt, soronként 80 karakterből vagy karakterhelyből, és minden egyes karakterhez egy attribútumbájt is hozzátartozik, amely megmondja az adott karakter háttér- és írásszínét, valamint azt is, hogy villogjon-e vagy ne. A 640x480 képpontból álló grafikus képernyő Truecolor üzemmódban, ami ugyebár mintegy 16 millió színárnyalatot enged meg bitenként, vagyis 16 millió szorozva 640x60 bájttal, azaz megközelítőleg 2 Mbájt.

Ehhez már spéci grafikus gyorsítókártyákat alkalmaztak, amelyekre gyors video-RAM-okat integráltak. Tehát már

itt is a memória. Ha a képernyő tartalmát hűen, sőt még aprólékosabban meg akarjuk jeleníteni, a printerünkbe is több Mbájtnyi memóriabővítő modult kell tennünk. S ez mind a szép, egyenletes vonalvezetésért, a kisimult, nem lépcsőzetes görbületekért.

Memória. Cache-memória a processzorok kisegítésére, vagy más néven gyorsítótár, amelyet a proci egyszerűbben tud címezni, így egy cache-t vezérlő áramkör a processzor kezére játszik. Sokan szeretnék tesztelni cache-RAM-juk sebességét, de erre megfelelő tesztprogramot nem lehet írni. A cache vagy be van kapcsolva, és akkor növeli a feldolgozási sebességet, vagy ki van kapcsolva.

Ugye, milyen bonyolult? Pedig a RAM egy bitjét egy kétállapotú félvezetődarabka testesíti meg. Ráadásul ez a dinamikus RAM-ok esetén még csak nem is stabil állapot, így azután állandóan ki kell olvasgatni, hogy a kiolvasásra fordított energia újra erőt adjon a kis „szerencsétlen bitnek".

RAM-hiba. Míg rájövünk, mindenre gondulunk, csak erre nem. Megmagyarázhatatlan elszállások, lefagyások. Nyári napon túlmelegedések.

1 Mbájt RAM fajlagos ára mintegy 4000 Ft plusz áfa. Ami érdekes, hogy itt hiába vesz az ember nagyobb kiépítésű memóriamodulokat (4 Mbájt, 6 Mbájt vagy 16 Mbájt), nem lesz olcsóbb, mintha 1 Mbájtos modulokból szedte volna össze. Ráadásul az alaplap memóriafoglalatai alapvetően meghatározzák, hogy azokba milyen minimális méretű RAM-modulok dughatók. Egy átlagos gép, eredetileg mondjuk 8 Mbájt RAM-os, 40 000 Ft-nyi RAM-ot tartalmaz, a konfiguráció teljes árának majdnem 30%-át. Külföldön már külön rablóbandák specializálódtak a RAM-okra. Nem a számítógépeket viszik el, hanem csak a lelkét. A processzort és a RAM-ot. Merthogy kis helyen elférnek. Éppen ezért könnyen csempészhetők zsebben is. Erről igazán a Vám- és Pénzügyőrség tudna mesélni.

Egyre több RAM kell a PC-be. Egyes programok dobozán nagyon szerény minimumérték szokott szerepelni a RAM-igényt illetően, a javasolt (recommended) érték már közelebb van a valósághoz, de a tapasztalt felhasználó már jól tudja, hogy windowsos vagy Windows NT-s, OS/2-es, illetve Windows 95-ös programok esetén az ajánlott értéket néggyel megtoldva kapjuk meg az igazán üdvös RAM-mennyiséget. Hogy miért becsülik szándékosan alul a gyártók a RAM-igényt? Csupán azért, hogy a tisztelt vevő nehogy elriadjon az extrém nagy értékektől. Később majd úgyis rákényszerül, és rászokik a RAM-bővítésre.

Ha kevés a RAM, a virtuális memóriakezelésre képes operációs rendszerek a harddiszkből hasítanak számunkra látszólagos memóriaterületet, többnyire átmeneti tárolónak, de így is érezhetően lassul egy program, amennyiben merevlemezen leképzett virtuális memóriához kell folyamodnia. Lassan a RAM-ok mérete túlhaladja a 3 évvel ezelőtti harddiszkek átlagos kapacitását.

Ha valakinek kevés a RAM-ja, mégis mindenképpen el akar indítani memóriaéhes programokat, manapság elég sok virtuális RAM-bővítő áll a rendelkezésére. Ezek ma reneszánszukat élik: RAM Doubler for Windows, MagnaRAM 2, SoftRam for Windows 95, Hurricane stb. Ha rövid a kardod, toldd meg egy lépéssel. Ha rövid a lépésed, vegyél nagyobb cipőt. Legfeljebb kilépsz belőle.

Vajon van-e megállás? Az Intel már csak Pentium processzorokat gyárt. Ezek ára valószínűleg még 1995 végén 30 ezer forint alá fog esni. Csak azok a fránya RAM-árak nem mozdulnak el lefelé! Konklúzió: a legjobb számítástechnikai befektetés manapság a RAM-vásárlás...

Memória, memória. Nem jut már eszembe semmi. Baj van a memóriámmal. De ez sajnos egy régi konstrukció, és már nem bővíthető tovább...

Herczeg József

Kingston's recommended memory requirements

TYPE OF USER	TYPICAL USAGE	RECOMMENDED RAM
Light Administrative	Light word-processing, light e-mail.	8MB - 12MB
Medium Administrative	Word-processing, e-mail, fax & communications software, database. 1-2 applications open at a time.	12MB - 16MB
Heavy Administrative	Word-processing, e-mail, fax & communications software, spreadsheet, business graphics. >3 applications open at a time.	16MB - 24MB
Light Customer Service	E-mail, database. 1-2 applications open at a time.	8MB - 12MB
Heavy Customer Service	E-mail, database, fax & communications, word-processing. >3 applications open at a time.	12MB - 20MB
Light Number Crunching	Spreadsheet, e-mail, accounting software. 1-2 applications open at a time.	16MB - 24MB
Heavy Number Crunching	Spreadsheets, statistical applications, research, large databases. >3 applications open at a time.	24MB - 32MB
Executive Management	Word-processing, e-mail, fax & communication software, spreadsheets, business graphics programs, database, presentation software. >3 applications open at a time.	16MB - 32MB
Light Graphics	Word-processing, page layout software, illustration/graphics software. 1-2 applications open at a time.	32MB - 64MB
Medium Graphics	Word-processing, page layout software, illustration/graphics software, basic photo editing, presentation software, font packages, multimedia. >3 applications open at a time.	64MB -128MB
Heavy Graphics	Photo editing of high resolution scanned images. Memory requirements approximately 3 to 4x file size of each simultaneously open file.	128MB - 512MB
OPI Print Servers/RIPS	High to low resolution substitution print servers, color and film output devices.	64MB - 512MB
Light-Medium Design	CAD software, CAM software.	32MB - 128MB
Heavy Design	3D CAD software, CAM software, solid modeling, rendering.	256MB - 2GB

Memorizálás

Így látja a processzor

A processzor szemszögéből tekintve a memória olyan, mint egy jókora, sok-sok fiókkal ellátott szekrény. Minden fiókba egy bájt fér, és minden fióknak van egy azonosító száma, ez a bájt memóriacíme. Amikor egy memóriahelyre írni, vagy onnan olvasni akar, a memóriacímet küldi el a címbuszon. A cím a memóriában kiválasztja a megfelelő helyet, és írásnál az adatbuszon küldött adatot tárolja ott, olvasásnál pedig a fiók tartalmát az adatbuszon elküldi a CPU-nak.

Azt, hogy a CPU maximálisan mekkora (hány bájtot tároló) memóriát tud kezelni, a címbitjeinek a száma dönti el. A 386/486 és Pentium processzorok 32 bites címmel dolgoznak, ami maximálisan 4 gigabájt (GB) memória használatát teszi lehetővé.

A memóriának a cím kiküldése után időre van szüksége a kért írás/olvasás végrehajtáshoz. A processzor működését órajel ütemezi, és általában sokkal gyorsabb, mint a memória, ezért a memóriavezérlő egy Readynek nevezett jellel tudatja vele, hogy a memória befejezte a műveletet, ahogy azt az 1. ábra mutatja.

A memória azért különleges a CPU számára, mert adatokat ír/olvas a perifériákra(ról) is, de utasításokat csak innen hajlandó végrehajtani. Ezért kell a futtatáshoz minden programot a háttértárolókról (floppy, harddiszk, CD-ROM stb.) a memóriába tölteni. A kiírás-betöltés során a CPU a program utasításait adatként kezeli, végrehajtani csak a memóriából lehet őket.

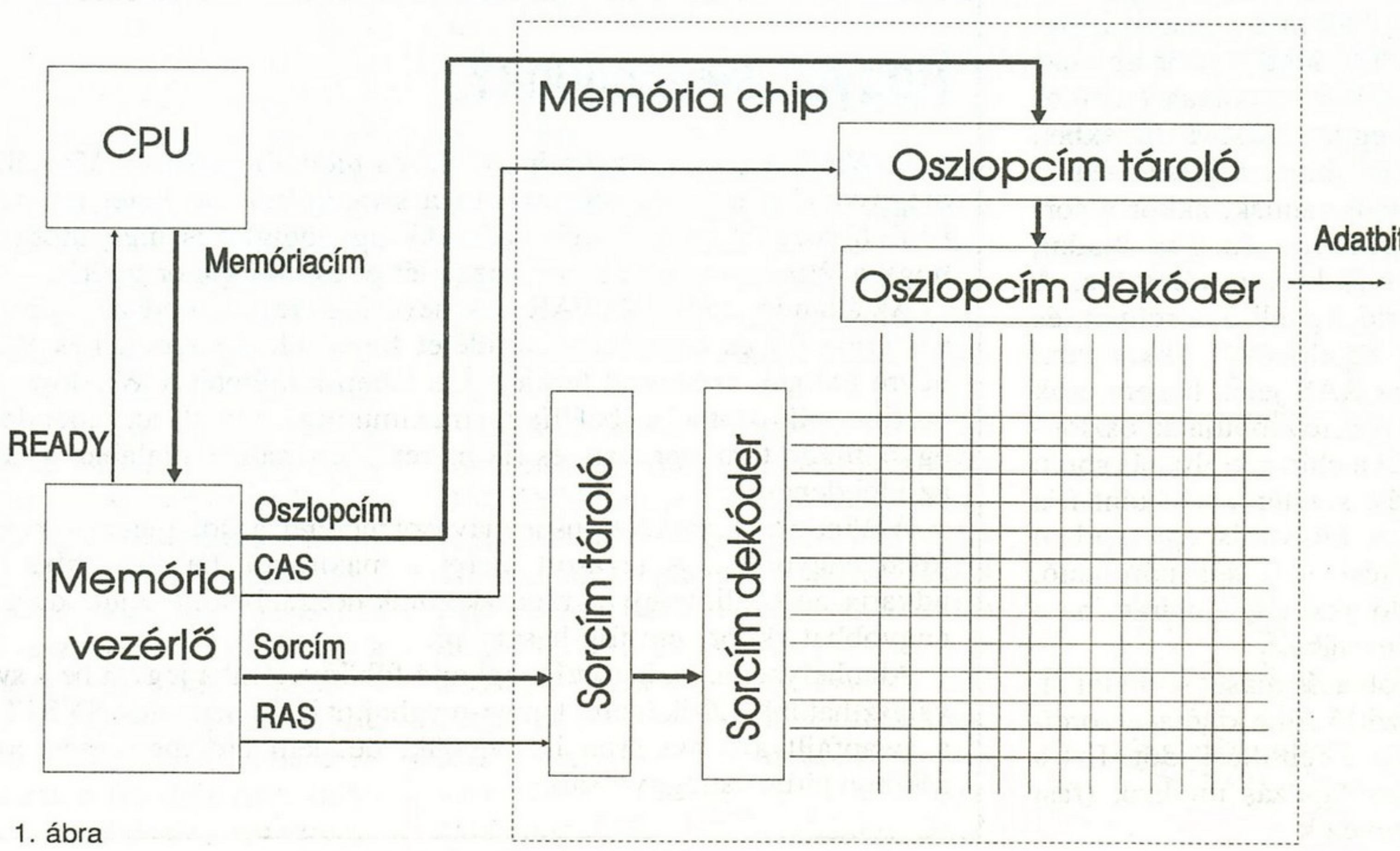
Ezért a processzor önmagában, memória nélkül működésképtelen. Amennyiben valamilyen szerkezetbe, pl. a billentyűzetházba pillantva csak egyetlen IC-t látunk, akkor egy integrált mikroszámítógéppel van dolgunk. Ebbe a CPU-n kívül beépítették a kiszolgáló áramköröket, és a működtető programot tartalmazó ROM/EPROM memóriát is. Az EPROM-ot tartalmazók felismerhetőek a tokon lévő kerek kvarcüveg ablakról.

A memória belülről

Az írható/olvasható memóriákat ma három csoport alkotja: a dinamikus RAM (DRAM), a statikus RAM (SRAM) és a flash memória. A PC-ben az operatív memória DRAM, ezért ezt vizsgáljuk meg részletesen. Az SRAM a cache-ben, míg a flash memória PCMCIA-kártyán fordul elő PC-környezetben.

Minden memória bitenként tárolja az információt, párhuzamosan több chippel alakítják ki a bájtot tároló modult.

A dinamikus RAM-ban a bitet egyetlen tranzisztor és egy kis kapacitás (kondenzátor) tárolja, míg a statikus RAM-ban egy bithez 4 vagy 6 tranzisztor kell. Ezért azonos technológiával a DRAM-ban az SRAM tároló kapacitásának a többszöröse valósítható meg. Ez a kis kapacitás azonban viszonylag rövid idő (10-15 ms) alatt elveszíti a töltését. Ha az információt meg akarjuk őrizni, ezt újra és újra pótolni kell,



1. ábra

szaknyelven frissíteni kell a memória tartalmát.

Az 1. ábrán látható egy DRAM memóriachip belső felépítése. A bitek mátrixba rendezettek, a megfelelő sor és oszlop kiválasztásával érhető el a kívánt bit.

A processzor erről azonban semmit sem tud, a memória kezelése a memóriavezérlő feladata. A CPU kiadja az elérni kívánt memóriahely címét a címbuszon. Ezt a címet a memóriavezérlő három részre bontja. A legalsó — legkisebb helyi értékű — rész lesz az oszlopcím, a középső a sorcím, a felső rész pedig az egyes memóriamodulok kiválasztását szolgálja. A cím pontos felosztása attól függ, hogy milyen kapacitású memóriamodul van az alaplapon.

A 2. ábra mutatja a memória címzésének a menetét. A memóriavezérlő először a sorcímet adja ki, amelyet a memória felé az RAS jellel érvényesít. A chip a sorcímmel kiválasztja a sort, ezután a vezérlő az oszlopcímet továbbítja a CAS jellel, amellyel a memória a sorból kiválasztja a megfelelő bitet. A műveletet 8 chipen végzi párhuzamosan, így áll össze egy bájt.

A memória tokján a hozzáférési időt szokták feltüntetni, ez a mai DRAM-oknál általában 70 ns (nanoszekundum). A kiolvasással azonban a kapacitás elveszíti a töltését, amit a DRAM automatikusan visszaír, de ehhez szükséges a feltöltési idő, ami általában azonos a hozzáférés idejével. Valójában tehát egy 70 ns-os DRAM-ból csak 140 ns-onként lehet kiolvasni egy-egy bájtot.

Szerencsére a helyzet nem ilyen súlyos, ez az idő többféle módon csökkenthető. A CPU legtöbbször egymás után következő bájtokat olvas a memóriából, ami az egyes DRAM chipekben egymást követő biteket jelent. Ha a bitek egy sorban vannak, akkor a sorcímet felesleges újra és újra kiadni, hiszen a sor már ki van választva. A memóriavezérlő figyeli a sorcímet, és ha az azonos az előzővel, akkor nem veszi vissza az RAS jelet, hanem csak a CAS állításával továbbítja az oszlopcímeket. Mivel a chip a kiolvasás során elveszett töltést szintén soronként írja vissza, amíg a kiolvasás egy sorban marad, a feltöltési idő is megspórolható. A feltöltési idő akkor is elmarad, ha a CPU ír a memóriába.

Amikor az olvasás másik sorra tér át, a memóriavezérlő újra kiadja a sorcímet, és kivárja a feltöltési időt. Ezt a módszert gyors lapozás módnak (fast page mode) nevezik.

A 32 bites adatbuszhoz 32 bit széles memória kell, ezért a 8 bites SIMM modulokból 4 db alkot egy egységet, az ún. bankot. Ez viszonylag sok helyet foglal az alaplapon, ezért terjednek a 32 bites SIMM modulok, amelyekből egy-egy darab egy bank. Két feltöltött bank esetén — 2 darab 32 bites vagy 8 darab 8 bites modul — alkalmazható az átlapolás (interleave mode), amikor a processzor felváltva fordul a bankokhoz. Ez azt jelenti, hogy az egyik bankhoz fordulás alatt a másik regenerálódik, vagyis nem jelent kiesést a feltöltődési idő.

Ha meghibásodik a memória

A memória hibája komoly veszélyforrás, hiszen egy téves utasítás vagy hibás adat komoly adatvesztést okozhat a gépben.

A legegyszerűbb — a soros adatátvitelben is alkalmazott — hibaellenőrzési módszer a paritásbit alkalmazása. A memória a bájtot alkotó 8 adatbit mellett egy paritásbitet is tárol. A paritásbitet egy, a processzortól független logika generálja és ellenőrzi. Amikor a CPU adatot ír a memóriába, a logika megszámolja az adatbájtban lévő egyeseket. Amennyiben a számuk páros, akkor a 0-s paritásbitet generál, ha páratlan, akkor 1-est, ami a bájttal együtt bekerül a memóriába. Az adat visszaolvasásakor ismét megszámolja az egyeseket, a paritásbittel együtt. Ha páros, akkor minden rendben, ha azonban páratlan, akkor valamelyik bit hibás. Ezt egy megszakításon keresztül jelzi a CPU-nak, amely egy hibaüzenet képernyőre írása után — „Parity error" vagy valami hasonló — leállítja a rendszert.

A fenti példa páros paritás, mivel az egyesek számát — ha szükséges — a paritásbit párosra egészíti ki. A módszer lényegén nem változtat, ha az egyesek száma mindig páratlan, azaz páratlan paritást használunk.

A memória és az alaplap technológiája sokat fejlődött az elmúlt években, a költségcsökkentés sok gyártót arra késztetett, hogy elhagyja a paritásbitet a memóriamodulról. Ezért a legtöbb BIOS setupban kikapcsolható a paritásgenerálás és -ellenőrzés. A másik oldalon viszont egyre több memóriát használunk egy gépben, ami növeli a hiba előfordulásának esélyét.

A paritás csak egyetlen bit hibáját jelzi, és az sem azonosítható a bájton belül, ezért nincs jobb megoldás, mint a leállítás.

A kritikus rendszerek azonban nem állíthatóak le egyik pillanatról a másikra, ezért itt ellenőrző és hibajavító ECC kódot (ECC = error checking and correcting) alkalmaznak.

A memóriavezérlő ekkor a memóriába írt minden bájthoz előállít egy bitkombinációt, amelyet az adatbájttal együtt tárol. Az adat kiolvasásakor ismét előállítja az ECC biteket, és összehasonlítja a memóriából kiolvasottal. Ha egyezik, minden rendben, ha nem, valamilyen hiba történt. Az ECC együtt használható a paritásbittel, és az ECC kód a hosszától függően — ami 5-8 bit között lehet — több hibás adatbitet is képes észlelni, egybitnyi hibát pedig korrigálni. Így egy adatbit meg-

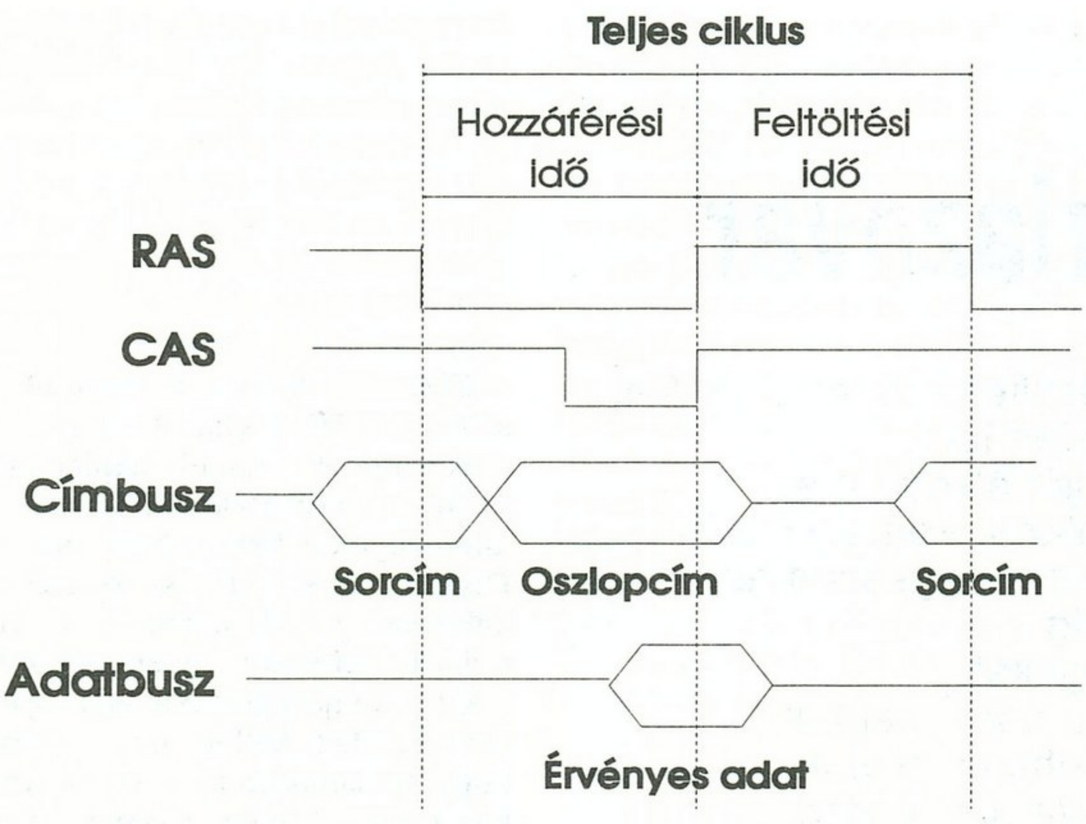
Hogyan swapoljunk?

A Windows 3.x a Vezérlőpult/386-os mód ablakában a Virtuális memória megnyomása után ad információt a swapfájlról, ott lehet megváltoztatni a beállításait is. A swapfájl lehet állandó vagy ideiglenes, mely utóbbi azt jelenti, hogy a Windows indításakor hozza létre, és kilépéskor letörli.

Az állandót 386PART.PAR-nak nevezi, és rejtett rendszerfájlnak állítja be, a mérete fix és összefüggő területet foglal el. Az ideiglenes Win386.SWP névre hallgat, széttagolt területen is lehet, a méretét a Windows az igények szerint változtatja a beállított maximumig. Mivel az állandó swap-fájl egybefüggő területen van, és fix méretű, használata általában gyorsabb, mint az ideiglenesé.

A Windows mindkét típushoz javasol méretet, az ideigleneshez az állandónál jóval nagyobbat. A javasolt méret a maximum, ha nagyobbat választunk, udvariasan közli, hogy ha ragaszkodunk hozzá, beállíthatjuk, de a javasoltnál nagyobbat akkor sem fog használni.

Alaphelyzetben a betöltő meghajtó főkönyvtárába jegyzi be a swapfájlt, de választhatunk másik merevlemez-meghajtót is. Itt nem, de a SYSTEM.INI-ben a swapfájlt alkönyvtárba is tehetjük, de nem érdemes, mert az elérése a főkönyvtárból a leggyorsabb.



2. ábra

hibásodása miatt még nem kell azonnal leállítani a rendszert, a hibát pedig jelzi a rendszerkezelőnek.

Az ECC kód miatt természetesen kevesebb adat fér a memóriába, az ECC generáló és ellenőrző beépítése pedig megdrágítja az alaplapot.

Frissíteni kell

A dinamikus RAM felépítéséből adódóan 10-15 ms után elveszíti a tartalmát, azt újraírni, azaz frissíteni kell. Szerencsére ez nem jelenti azt, hogy a processzornak kell minden adatot kiolvasnia és újraírnia, mert ekkor a memória mennyiségének növekedésével a hasznos munkára alig maradna ideje.

A frissítést külön egység végzi az alaplapon, és a memóriachip soronként frissít, tehát elegendő a sorcímeket szabályos időközönként végigpörgetni. A frissítést közvetlenül a gép bekapcsolása után el kell kezdeni, a frissítő logika ekkor indul a nulladik sortól. Az alaplapon lévő időzítő 15 mikroszekundumonként ad parancsot a frissítésre. A frissítő logika elkéri a buszt a processzortól, és az éppen következő sorcím kiadásával egyidejűleg minden memóriamodulban frissít. Ezután visszaadja a buszt a processzornak, hogy újabb 15 mikroszekundum múlva a következő sor frissítésére kérje el.

A régebbi, 286-os rendszernél pontosan kiszámítható volt, hogy a frissítés mekkora időveszteség a CPU, és így a gép teljesítményének szempontjából. A 486-osok és a Pentium mind belső, mind külső cache-sel rendelkezik, amelyek használatát a frissítés nem befolyásolja, ezért az időkiesés nem számítható, az esetek 99%-ában ha a cache elég nagy, a CPU észre sem veszi.

A mai DRAM-oknál a sorcím kiadására sincs szükség, elegendő megfelelő időközönként az RAS jelet a CAS visszavonása után egy ideig még fenntartani, a chip önmagától frissít. Ezt nevezik rejtett (hidden) frissítésnek, ami a BIOS setupban beállítható.

Az operációs rendszer szerint

Az átlagos felhasználónak a memóriára hosszú ideig nem kellett sok figyelmet fordítania. A RAM 640 KB volt, amiből 40-60 KB-ot elfoglalt a DOS, a maradékot használhatta az éppen futó program. A korláttal minden programozó és fordító tisztában volt, ritkán fordult elő a „Kevés a memória" hibaüzenet, és a probléma néhány rezidens program elhagyásával akkor is megoldható volt.

A memóriára a Windows elterjedése óta kell jobban odafigyelni. Gyakori a „Nincs elég memória" hibaüzenet, és a kevés memóriával rendelkező gépek egy újabb program vagy nagyobb adatfájl betöltése után elképesztően lelassulnak.

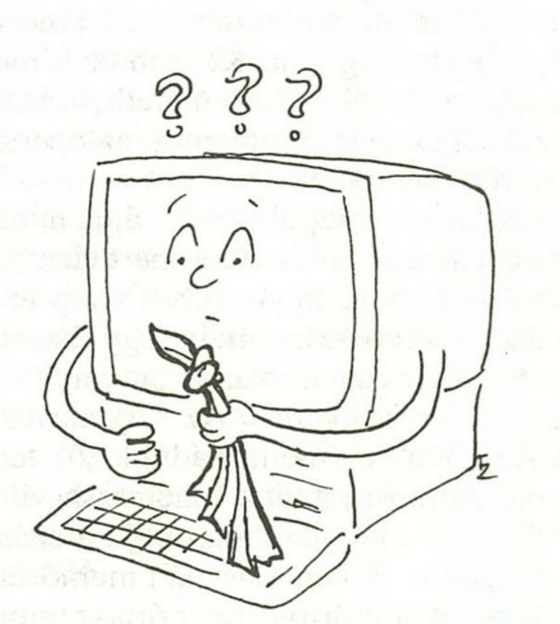
Tipikus eset: a felhasználó Windows alatt lassúnak érezve a gépét (egyébként DX2 66 MHz CPU-val) processzorcsere mellett dönt, és egy DX4-100 MHz-est tesz az alaplapra. A memóriája 4 MB, ezt nem bővíti, egyrészt mert drága (a DX4 12-13 ezer forint plusz áfa, és a DX2-t beszámítják, az újabb 4 MB RAM pedig 16 ezer plusz áfa), másrészt, mert a szoftver dobozán azt olvasta, hogy 2-4 MB elegendő a használatához. Amíg csak egy-két kisebb programot (Write, Paintbrush stb.) használ, jól érezhető a sebességnövekedés. Amint azonban betölti az Excelt és a Wordöt, gépe ugyanúgy lelassul, mint korábban. Emberünk dühöng, szidja a Windowst, a processzort, a hardverkereskedőt.

A Windows egy multitaszkos, egyidejűleg több program futtatására alkalmas operációs rendszer, a több programnak, a többletszolgáltatásnak, a grafikus felületnek pedig memória kell. Azért, hogy a gépben lévő RAM memória mennyisége ne korlátozza közvetlenül a futó programokat, a betölthető adatokat, a Windows is bevezette a virtuális memóriakezelést. Ennek megvalósítása a CPU szintjén elég bonyolult, a gyakorlatban azonban egyszerűen azt jelenti, hogy ha nincs elég fizikai memória, a Windows az éppen nem használt részeket kimásolja a merevlemezre egy ún. swapfájlba, hogy helyet szabadítson fel a memóriában. Persze ha a kimásolt részt később használni akarjuk, vissza kell tölteni. Ekkor egy másik memóriarészt tesz ki, vagyis csereberél („szaknyelven" swapol) a memória és a merevlemez között.

A merevlemez sebessége azonban nagyságrendekkel kisebb a RAM sebességénél. A swapfájl gyakori használata nagyon lelassítja a rendszert, hiszen a Windows egyrészt nem tudja, hogy mit fogunk használni legközelebb, másrészt pedig a szokásos lemezműveleteinket — mentés, betöltés — is el kell végeznie. Ezért a Windows alatt egy program elindulása nem feltétlenül azonos a tempós munkával. Ráadásul a Windows szépíti a helyzetet, mert a Névjegy-ben kijelzett szabad memóriába beleérti a swapfájlt is.

A felhasználáshoz képest kevés memória törvényszerűen lelassítja a rendszert, fenti példában a CPU-cserének — önmagában — nem sok értelme volt.

Csórián Sándor



A nagyok kis stiklijei

Tudatos memóriazavar

Annak idején, amikor Bill Gates elkészítette az első DOS-t, talán maga sem gondolta, hogy minden idők legszerencsétlenebb memóriamodelljét alkotta meg. A későbbiek során, talán az üzlet, talán a kompatibilitás jegyében már nemigen változtattak ezen, egészen az ún. védett (protected) mód megjelenéséig. Sőt, egyszer itt is meggondolták magukat. Az Intel a 286-os processzorban már implementált valamilyen védett módot, de ennek kihasználását egy-két amatőr szoftverírótól eltekintve akkor még senki sem látta piacilag érettnek.

A memóriaméret nőtt. A kezdeti gépek „hatalmas" — 8 KB-os — memóriája után megjelentek a DOS-világban a 256, majd az 512, végül a 640 KB-os gépek. A gond pedig akkor kezdődött, amikor az operatív memória túllépte a 640 KB-os elméleti határt, és az a kezdetben hiányzó merevlemezen a szerencsétlenül választott FAT-struktúra határait feszegette. Először 20 MB, majd 40, 256, napjainkra pedig 450 MB az elméleti korlát. E felett adott a lehetőség a trükközésre, ami újabb piacokat teremt a szoftverforgalmazóknak, nem beszélve az egymással való kiszúrás, a beépített inkompatibilitás lehetőségéről.

A DOS 640 KB-os memóriamodelljét az XMS (Extended Memory System), valamint az EMS (Expanded Memory System) technológia lépte át először. Itt a trükk abban állt, hogy a 640 KB-os operatív tár egy adott területére segédprogramokkal belapozzuk azt a felső memóriaszegmenst, amelyre éppen szükség van. Később ez a megoldás a DOS foltja maradt, azaz a szükséges segédprogramok az operatív tár részévé váltak.

A másik megoldás az éppen nem használt memóriaterület merevlemezre való kiírása, az úgynevezett swap-technika (cserebere), amely egy hasonló programozástechnika, az overlay (átfedés) édestestvére. Az úgynevezett RAM-double (memóriaduplázó) technológia most jött át a Macintosh-világból a HP nyomtatókba, majd a számítógépekbe. A nem használt memóriaterületeket tömörítve memóriaszegmensekben tárolják, és onnan lapozzák be, illetve ami már ide sem fér, azt kiteszik a merevlemezre. Ez utóbbi azonban jelentős lassulást eredményez, igaz, kevesebb a lemezigénye, mint a swapolásnak.

A Microsoft által használt megoldás már tömegtermelési kellék volt, a szokásos bug-hegyekkel, problémákkal. Ezt a piaci szituációt kezdték a kívülálló gyártók is felismerni. Két törekvés jelentkezett. Az egyik gyártó azt mondotta — Digital Research: béke poraira! —, hogy az egész DOS-t újra kell írni. Mégpedig úgy, hogy megőrizzük a meglévő programokkal a kompatibilitást, de a korlátokat le kell dönteni. Azaz azokat a memóriakezelési ötleteket, amelyeket később más cégek külön programokban megvalósítottak, magának az operációs rendszernek a részeként kell megírni.

A dolog egy ígéretes termékkel lepte meg a piacot, a DR-DOS-szal. Ezt immár az üzlet védelmében Bill Gates és csapata nem nézhette jó szemmel, hiszen ők akkor már úton voltak a csúcsra. Éppen ezért, amikor megjelent az első Windows program, elérkezettnek látták az időt a visszavágásra. A Windows valami rejtélyes okból nem futott DR-DOS környezetben. Állítólag volt benne egy kódrészlet, amely bevezette a „generális faliórát", minden Windows-felhasználó rémét (General Protection Failure = általános védelmi hiba). Erre szerencsétlen felhasználók csapatai rohanták meg a Digital Researchöt, hogy ez vagy az a program nem megy a rendszerükön.

Elkezdődött a patch-háború (a patch a verzióközi javítások elnevezése a szakzsargonban), mely napjainkra egyre jobban eszkalálódik. Először olyan állományokat kellett kifejleszteniük a Digital Research szoftverészeinek, amelyekkel a Windows egyes állományait lecserélve az operációs rendszerük fut, majd később a saját operációs rendszerüket kellett úgy módosítani, hogy fussanak rajta a Windows-alapú programok. Mindez a cég elvérzéséhez vezetett. A Novell becsületére legyen mondva, hogy Novell DOS néven tovább vitte az ügyet, de ugyancsak hasonló nyűgökkel, majd a Microsoft és a Novell közötti egyezség vetett véget az áldatlan állapotnak: a megoldás természetesen csak az lehetett, hogy a Novell leáll a Novell DOS fejlesztésével. Így bár az a DOS már sokkal jobb volt, mint a Microsofté, mind a mai napig kénytelenek újabb javításokat kiadni annak érdekében, hogy a korábbi vásárlók érdekei ne sérüljenek.

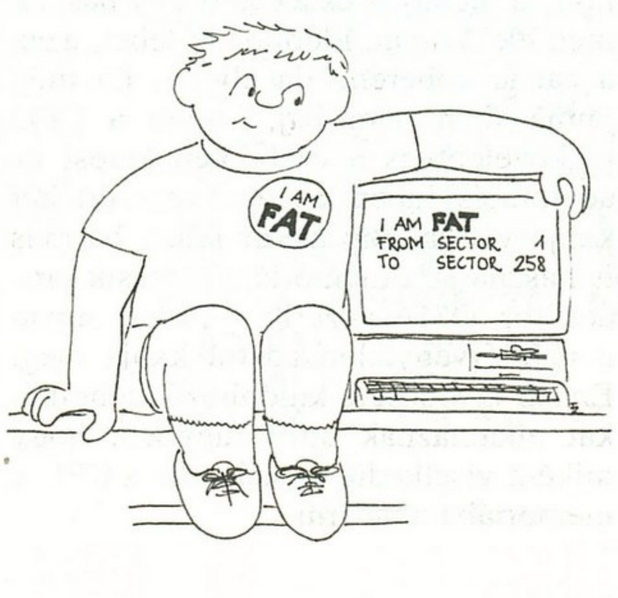
A másik konkurens, az orosz PTS fejlesztőcsoport által készített DOS más utat választott. Ők nem az eredeti DOS megoldásait, hanem annak szoftveres felületeit használják fel, azaz újraírták az egész rendszert. Éppen ezért mindig egy-két lépéssel hátrább vannak. Ugyanakkor a programban két megoldást választanak maguknak. Az egyik a Windows-kompatibilis üzemmód, amikor teljesen a hagyományos DOS-felületet kapjuk, annak minden hibájával. A másik saját üzemmódban viszont jóval egyszerűbben és gyorsabban fut minden olyan program, amelyet tisztán programoztak. Azaz a program írói nem trükköztek, hanem a publikált és szabványos rendszerhívásokat és felületeket használták. Sajnos a mai világban egyre kevesebb az így készült szoftver...

A történet azonban nem lenne kerek, ha nem szerepelnének benne azok a gépgyártók, amelyeknek volt egyéni elképzelésük a memóriamenedzselésről. Ebben a műfajban élen járt a Compaq. Olyannyira, hogy a Microsofttal kötött megállapodása alapján forráskódszinten revideálhatta a DOS egyes dolgait. Így született meg az OEM jellegű Compaq DOS és annak az 5.0-s verziója, amely a Tandon cég hasonló

OEM-változatának a nyomába lépett. A Tandon DOS 3.31 volt az első, amely 250 MB-ig egy partícióban tudta kezelni a merevlemezt. A Compaq ezt átvette, majd elkezdte egyéni dolgokkal bővíteni rendszerét. Így állt elő az, hogy a Compaq gépek a sajátos memóriamenedzsment miatt csak saját DOS-szal futottak jól, minden más, a szabványoknak megfelelő gép viszont kifogástalanul futott a kipoloskázott Compaq DOS-szal.

Az Olivetti által választott út — hogy nem vették figyelembe ezeket a sajátosságokat — katasztrofális eredményre vezetett. Hiába adtak ki Olivetti DOS-verziót, a programok memóriakezelési hibával szálltak el az örök vadászmezőkre, általában adatainkat is magukkal vive. Így számukra nem maradt napjainkra más választás, mint beállni a sorba. Hasonlóan jó tanítványnak bizonyult a Hewlett-Packard is, mert gépei eljutottak az egyénieskedéstől a Microsoft-kompatibilitásig. Manapság lehet szidni vagy szeretni az operációs rendszerek készítőit, de az egész számítástechnikai világ az ő kezükben van.

A szerzői jogi törvény módosítása, a borítékszerződések általánossá válása a felhasználók, de még a programírók kiszolgáltatottságát is csak növelte. Erre utal a Microsoft és a Stack Corporation közötti zavaros bírósági vita. Itt a vita tárgyát egy nem publikált, de a DOS által használt, és a Stack által visszafejtett memóriamodell felhasználása szolgáltatta. Az amerikai bíróságot az az érvelés sem hatotta meg, hogy a nem dokumentált dolgokat az emberek a saját jól felfogott érdekükben fejtik vissza, illetve az operációs rendszert íróknak minden olyan tényt és adatot nyilvánosságra kell hozniuk, amely a vele kapcsolatos fejlesztőmunkához szükséges. Nem véletlen, hogy az USA-ban jogi kutatásokat folytatnak abból a célból, hogy szétválaszthatók legyenek az operációs rendszert író-forgalmazó



cégek a szoftverfejlesztőktől — a rendszerírók ne juthassanak piaci előnyhöz. Az inkompatibilitás oka ilyen esetekben mindig memóriazavar. Vagy talán az informatikai társadalom tartós elmezavara?

Az élet összehozhat hibátlan rendszerkombinációkat is. Az IBM szintén megjelent operációs rendszerrel a személyi számítógépek piacán. Ez különböző titkolt vagy bevallott licencvásárlások alapján a Microsoft adott DOS-verziójának belső és külső sajátosságait tükrözte. Újdonság volt a REXX makrónyelv, amellyel a Unix shell scripthez hasonló rendszerfüggetlen programokat lehetett írni. De a memóriamenedzserek is hasonlóak voltak.

A Windows és a DOS DPMI — azaz DOS Protected Mode Interface — jelentős változást hozott. Ez a változás szükséges volt, hiszen a külső gyártók egyre jobb DOS-extenderekkel (Phar Lap, DOS4GW stb.) jöttek ki, amelyek lehetővé tették egyre nagyobb és bonyolultabb programok futtatását, és a hagyományos memóriamenedzsment korlátainak látszólagos áthágását. Ezekkel a Microsoft többé-kevésbé megbékélt. Kevésbé tetszett viszont a Quarterdeck törekvése, amely multitaszkos környezetet szeretne DOS alatt biztosítani.

Folyik a harc: ki milyen inkompatibilitást épít be a másik programjával szemben szoftvereibe. Így a QEMM és társai lassan már több erőt kénytelenek a szoftverek javítására fordítani, mint magára a programfejlesztésre. Ugyanakkor ezzel programjaik konfigurálása egyre nehezebbé válik, ami a kifejezettem majombiztosnak szánt Microsoft programokkal ellentétben ár- és felhasználói hátrányt jelent. Így sajnos a paletta elszürkülésének vagyunk tanúi.

A Microsoft a memóriafelhasználással mindig is pazarlóan bánt. Nem izgatja, hogy a nagy memóriaigénnyel a hardvergyártók üzletét segíti. (Vagy nagyon is izgatja?) Néhány kézi számítógépre megírt és shareware program mutatja, hogy igényesen, kis memóriaigénnyel a mai bigware monstrumok funkcionális megfelelőjét milyen jól el lehet készíteni.

Eddig az operációs rendszer gépközeli valami volt. A Windows NT és a Windows 95 fejlesztése viszont arra mutat, hogy az operációs rendszert kezdik jól elhatárolható rétegekre bontani. Itt azután már felvetődik, hogy a kompatibilitást meg lehet-e őrizni a korábbi szoftverekkel. Egy kicsit a Novell 4.xx és a UnixWare filozófiája fertőzte meg a fejlesztőket. Ennél a koncepciónál az induló memóriaigény is nagyságrendekkel nagyobb, mint a hagyományos. Gépközeli szinten fut az operációs rendszer első köre, a kernel (mag). E köré épülnek az egyéb külső funkciók. Végül a nem rendszerközeli programok ún. virtuális gépen futnak, egymástól gyakorlatilag függetlenül. Az egymás közti kapcsolatot a rendszerközeli rész oldja meg, miként az erőforrások elosztását is. Ez az erőforrás-koncepció jelentősen megváltoztatta a programírást, és ha a Novell és a Microsoft folyamatos leszoktatási törekvései eredménnyel járnak, a hagyományos operációs rendszerekről áttérünk ezekre az új, kernelorientált rendszerekre.

Ezek előnye, hogy a programok egymástól függetlenül képesek futni, és az egyik programban esetleg fellépő hiba nem befolyásolja a többi futását. Itt a memóriakezelés is a kernel hatáskörében van, az osztja ki a pillanatnyi igények szerint, mint bármely más erőforrást. Ennek azonban ára van, a kompatibilitás részleges, majd teljes feladására kényszerülnek. Ez viszont a szoftverforgalmazók bevételeit növeli, mert egyre újabb piacok nyílnak meg. Ugyanakkor szuperhatalmat ad maguknak az operációs rendszert íróknak is, hiszen az általuk rendelkezésre bocsátott információk mennyiségétől és árától, valamint a fejlesztőeszközök milyenségétől függ, hogy egy külső fejlesztő mennyire hatásos és milyen árú szoftverrel tud kirukkolni a piacra.

A másik zavaró körülmény a hardvergyártók környezetében keresendő. Ugyanis a gépek memóriát igényelnek. Memóriát viszont csak néhány cég gyárt, miként processzort is. Miközben azonban a konkurens termékek jótékonyan hatottak az Intel processzorok árszintjére, ugyanez nem mondható el a memóriamodulok gyártóira. A gépek konstrukciós változásai ugyanis állandóan megkövetelik az újabb RAM-típusok megjelenését. Ugyanakkor a „hamis paritásos”, azaz paritásgenerátoros RAM-konstrukciók a gyenge minőségű terméket is eladhatóvá teszik. Olyan nagy a hiány — amelyről többször is bebizonyosodott, hogy tudatos piaci manipuláció eredménye —, hogy az árak minden racionalitással ellentétben stabilak, sőt nőnek. Vagy azért, mert valamire hivatkozva kevesebb terméket hoznak forgalomba a gyárak. Vagy azért, mert a hardvergyártók áttérnek egy új típusra. Vagy azért, mert az adott típust már nem gyártják. És ebben a manipulációban a vesztes mindig a felhasználó.

Kis János

Miért nem statikus ...?

Egy kis cache-ológia

A félvezető-technika gyors ütemű fejlődésével egyre több tranzisztor integrálható a szilíciumchipek felületére. Ez egyre bonyolultabb, és így egyre gyorsabb processzorokat eredményezett. A memóriánál a még több tranzisztor a tárolókapacitás növekedését tette lehetővé, de mivel a belső szerkezet alapjaiban nem változott, a sebesség egyre jobban elmaradt a processzoré mögött.

A ma általános 70 ns-os DRAM jelentős időveszteség nélkül csak a 386DX-33 MHz-es processzorig képes kiszolgálni a CPU-t. Ezért a későbbi vagy magasabb órajelű processzorokhoz cache memória szükséges. És a cache nemcsak az alaplapon jelent meg, hanem a 486-os óta az Intelével kompatibilis CPU-k is tartalmaznak belső cache-t.

A cache memória azért lehet gyorsabb, mint a dinamikus RAM (DRAM), mert statikus, vagyis SRAM-ból alakítják ki, aminek az elérési ideje tipikusan 15-20 ns. Miért nem SRAM-ból van az egész memória?

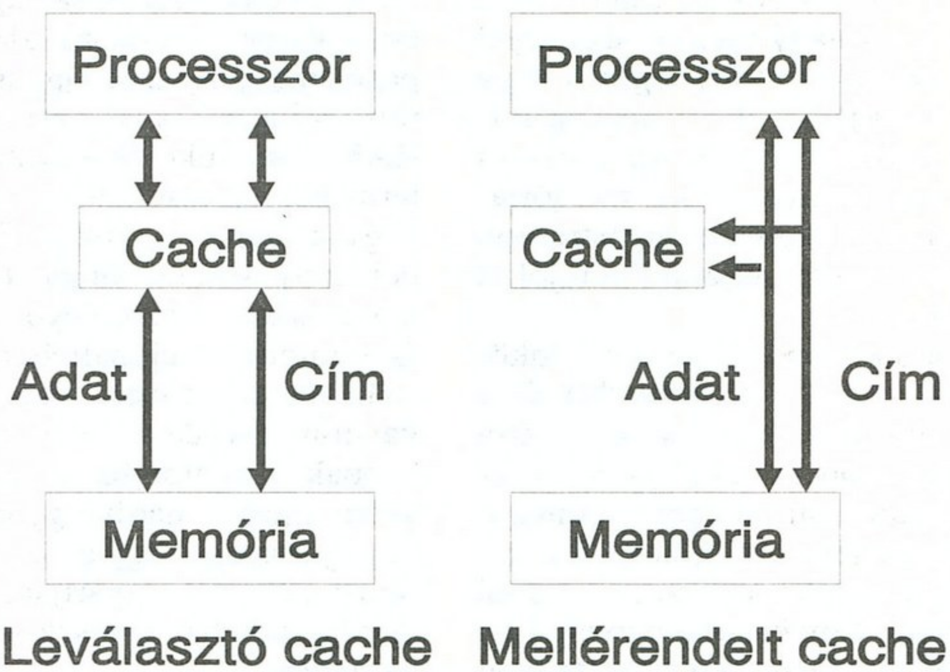
Először is az ára többszöröse a DRAM árának. Az egy chipen elérhető tárolókapacitás csak kb. a harmada, és a fogyasztása is lényegesen magasabb (gondoljunk a hordozható gépekre!). A cache technológia igen elterjedt mind a CPU-n belül, mind pedig kívül.

Milyen cache?

A cache kétféleképpen helyezkedhet el a processzor és a memória között, ahogy azt az 1. ábra mutatja. A *leválasztó (look-through) cache* vezérlője teljesen elválasztja a processzort a memóriától; ha a kért adat nincs meg a cache-ben, akkor a cache-vezérlő továbbítja a címet a memóriának. Mivel a cache-vezérlőnek időre van szüksége ahhoz, hogy megállapítsa, benne van-e a kért adat a cache-ben, ez ilyenkor több időbe kerül, mintha a CPU eleve a memóriához fordult volna.

A módszer előnye, hogy amíg a cache-ben megtalálható a processzor által kért információ, addig nem foglalja a memóriát, ami a multiprocesszoros, osztott memóriájú rendszerekben előnyös.

A *mellérendelt (look-aside) cache* tulajdonképpen párhuzamosan működik a memóriával. Amikor a processzor a cím kiadásával kéri az adatot, mind a memória, mind a cache reagál. Ha az adat megtalálható a cache-ben, akkor az egyszerűen megelőzi a memóriát. Egyébként pedig a memória a szokott módon szolgáltatja az adatot, nincs plusz késés, mint az első megoldásnál. A mellérendelt cache kialakítása egyszerűbb és olcsóbb, mint a leválasztó cache-é, ezért a legtöbb alaplapon ezt alkalmazzák. Egyetlen hátránya, hogy a processzor akkor is foglalja a memóriát, ha az adat végül a cache-ből érkezik, ezért multiprocesszoros gépekben nem használható.



1. ábra

Olvas és ír

Hogyan kerül az adat a memóriából a cache-be? A legtöbb esetben a cache egyszerűen tárolja a memóriából a processzor által kiolvasott információt, abból a megfontolásból, hogy arra a CPU-nak esetleg újból szüksége lesz. Olyan programkód esetén, amely soha nem kéri még egyszer ugyanazt az információt, ez a cache semmiféle előnyt nem jelent.

Szerencsére a valódi programokban számtalan ciklus fordul elő, és ismételten kellenek az adatok.

A leválasztó cache esetében, ha a memóriához kell fordulni, mert a kért adat nincs a cache-ben, a cache-vezérlő a CPU által kért bájt(ok) után következőket is beolvassa a cache-be — arra számítva, hogy a processzor előbb-utóbb ezeket is kérni fogja.

Egyszerű lenne a cache-tervezők dolga, ha a CPU csak olvasna a memóriából. Néha azonban ír is, ami azzal jár, hogy a memória és a cache egymásnak megfelelő adata különböző lehet, azaz a cache koherenciája elvész. Ez még önmagában nem baj, hiszen a CPU legközelebb is a cache-ben keresi az adatot, és innen az érvényes értéket kapja vissza. Baj akkor lehet, ha más is használja a memóriát — másik processzor, DMA-vezérlő —, mert onnan a már érvénytelen adatot kapja meg. Ennek kivédésére különböző technikákat alkamaznak attól függően, hogy miként viselkedik a cache, ha a CPU a memóriába akar írni.

Az *átíró (write-through) cache*-nél az írás mind a cache-ben, mind a memóriában azonnal megtörténik. Az írási műveletet ekkor a cache nem gyorsítja, de ha más — például a DMA-vezérlő — nem ír a memóriába, a cache mindig koherens marad.

A *visszaíró (write-back) cache*-nél a processzor csak a cache-be írja az adatot, így a cache az írást is gyorsítja. Ezután a cache és a memória tartalma különbözik. A cache-vezérlő csak akkor írja ki a módosított adatot a memóriába, ha kell a hely a cache-ben, vagy ha azt észleli, hogy egy másik CPU épp a módosított adatot kéri a memóriától. Ehhez persze folyamatosan figyelnie kell a memória használatát is, ami jelentősen növeli a cache-vezérlő bonyolultságát.

A két módszer közötti megoldás a késleltetett írás (buffered write vagy burst write). Az adat ekkor is csak a cache-be kerül, vagyis az írás is gyors, nem csak az olvasás, de a cache-vezérlő az első adandó alkalommal, amikor a CPU belső műveletet végez, kiírja a memóriába az adatot. Amíg ez meg nem történik, nem enged mást a memóriához férni, így nincs koherenciaprobléma.

Az alaplapon lévő cache írásmódját sok esetben mi is beállíthatjuk a ROM-BIOS setupban.

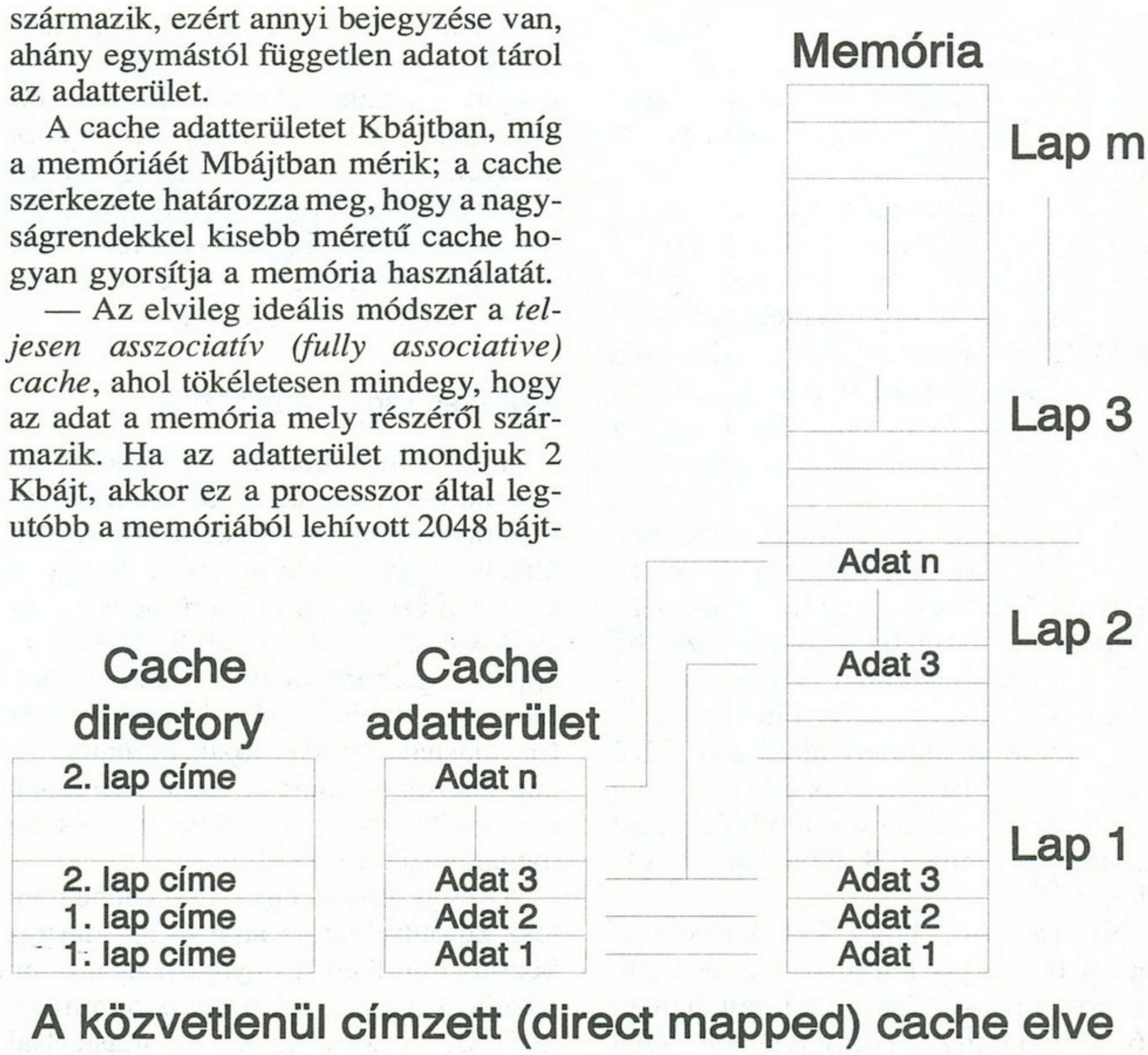
Cache-szerkezetek

A cache-vezérlőnek kell eldöntenie, hogy a processzor által kért adat benne van-e a cache-ben vagy sem. Ehhez pedig tudnia kell, hogy honnan, a memória melyik címéről származik a cache-ben lévő adat.

A cache-memória két részre oszlik, directoryra — hívják tag-RAM-nak is — és adatterületre. A directory tárolja, hogy az adat a memóriából honnan származik, ezért annyi bejegyzése van, ahány egymástól független adatot tárol az adatterület.

A cache adatterületet Kbájtban, míg a memóriáét Mbájtban mérik; a cache szerkezete határozza meg, hogy a nagyságrendekkel kisebb méretű cache hogyan gyorsítja a memória használatát.

— Az elvileg ideális módszer a *teljesen asszociatív (fully associative) cache*, ahol tökéletesen mindegy, hogy az adat a memória mely részéről származik. Ha az adatterület mondjuk 2 Kbájt, akkor ez a processzor által legutóbb a memóriából lehívott 2048 bájtnyi információt (adatot és utasítást) tartalmazza.



2. ábra

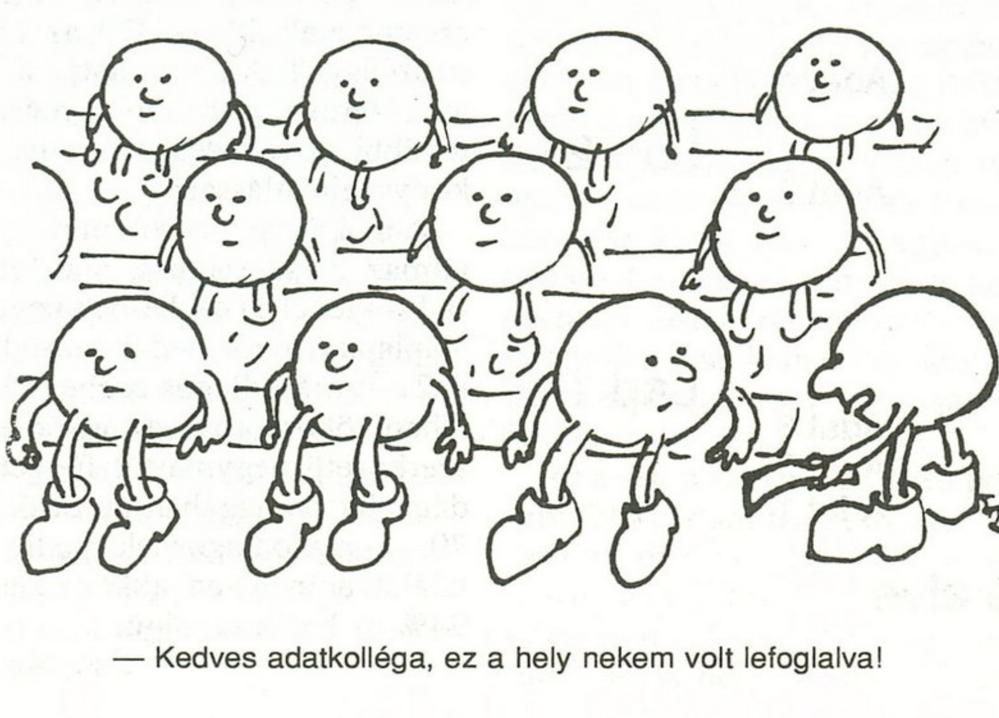
Az asszociatív cache-nek a legnagyobb hátránya, hogy mivel a cache bármely adata a memória bármely részéről származhat, a cache-vezérlőnek a teljes directoryt végig kell néznie, hogy megállapítsa: benne van-e az adat a cache-ben.

Például az adatterület 2 Kbájt, egy bejegyzés pedig 4 bájt (32 bit a 32 bites processzornak megfelelően). Ekkor a directoryban 2048/4 = 512 bejegyzés van. Ennek a végignézése olyan sok idő, hogy az a cache alkalmazását értelmetlenné teszi. Ezért vagy a bejegyzések — ezeket hívják más néven cache-soroknak vagy csak soroknak — méretét kell növelni, vagy pedig az adatterületet csökkenteni. Az asszociatív cache ezért általában kisméretű, 1-2 Kbájt, a directory pedig nem külön memória, hanem magának a cache-vezérlőnek a belső regiszterei, ami gyorsítja a keresést. Ilyen kis cache-méret a PC-ben azonban nem elegendő, ezért ebben a környezetben nem alkalmazzák.

— A PC-alaplapon a leggyakoribb a *közvetlen címzésű (direct-mapped) cache*. Ennél a megoldásnál, mint azt a 2. ábra mutatja, a memóriát a cache adatterületével megegyező méretű lapokra osztják.

A cache-ben az adat elhelyezkedése megegyezik a memórialapon belüli címével, ezért a directoryban elegendő a memórialap címét tárolni. Nézzünk erre is egy példát.

A cache adatterülete legyen 64 Kbájt, a memória pedig max. 256 Mbájt. Ekkor max. 256*1024/64 = 4096 darab 64 Kbájtos lapra osztja a memóriát. A processszor 32 bites memóriacímet ad ki, de mivel az alaplapon max. 256 Mbájt memória lehet, ezért csak az alsó 28 bitjét használja. A memórialapok



— Kedves adatkolléga, ez a hely nekem volt lefoglalva!

címe (0—4095) 12 biten tárolható, a lapon belüli cím pedig 16 bit. A 28 bites címből ezért a felső 12 a memórialap címe, az alsó 16 pedig a lapon belüli cím.

Amikor megkapja a CPU adatkérését, a cache-vezérlő az alsó 16 bittel külön-külön megcímezi a directoryt és az adatterületet. Ha a directoryból a cím felső 12 bitjét, azaz a lap címét kapja vissza, akkor az adatterületen erről a lapról származik az adat, így kiengedi a CPU-nak. Ha nem, akkor az adat nincs a cache-ben.

Amennyiben az adatterületen egy bájt lenne egy sor, akkor a directory nagyobb memóriát igényelne, mint az adatterület, hiszen 64 Kbájtnyi (azaz 65 532 db) 12 bites címet őrizne, míg az adatterület 8-szor ennyi bitet. Ezért a sor mérete a legtöbb alaplapon lévő cache-nél 16 bájt, vagyis példánkban a directory 65 532/16 = 4096 bejegyzést tartalmaz, amihez 8 Kbájt tag-RAM elegendő.

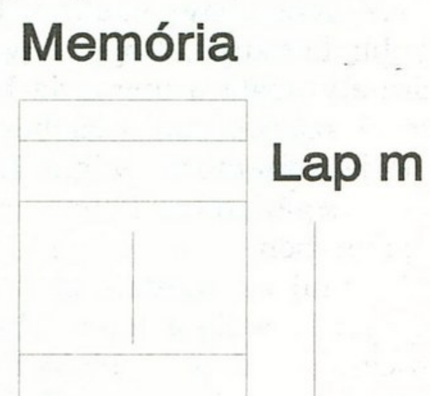
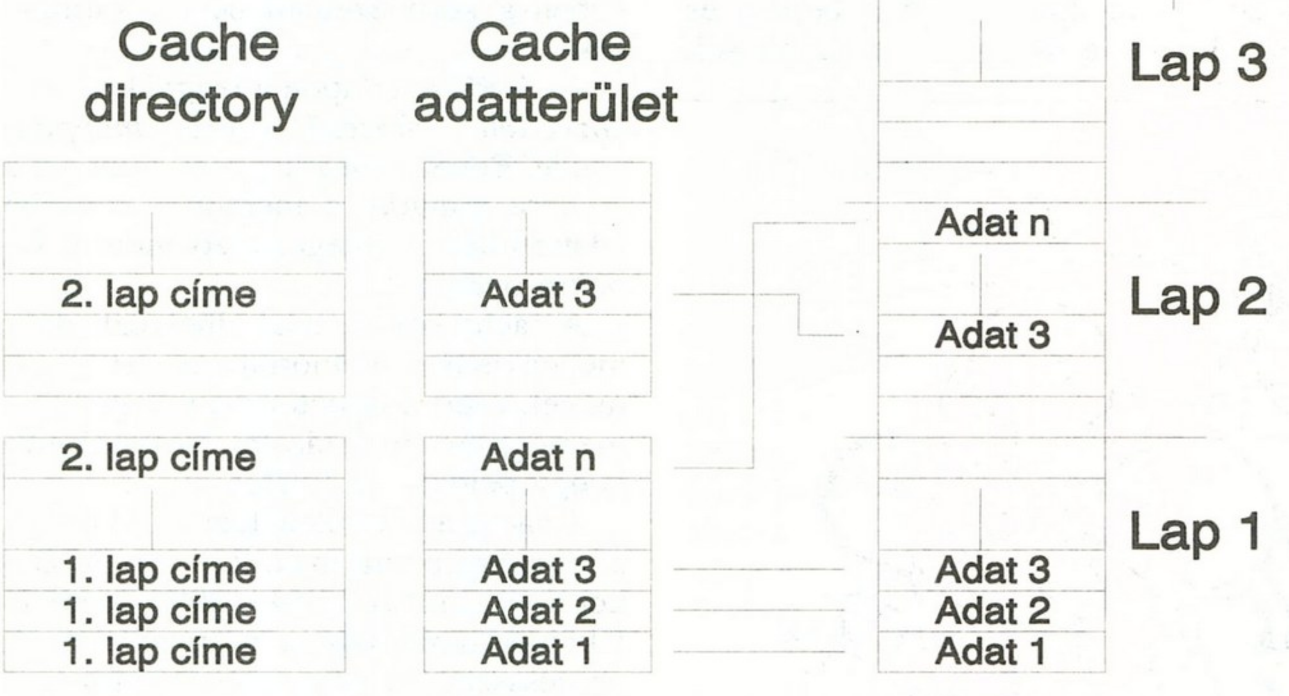
Nyilvánvaló, hogy két különböző lapról azonos pozíciójú — a lapon belül azonos című — adat (nem 1 bájt, hanem 16, hiszen ennyi a sorméret) nem lehet egyszerre a cache-ben. Az 2. ábrán a második lapról érkező Adat 3 felülírja az előzőt.

— Ezen segít a 3. ábrán látható *kétutas (two-way set) cache-szerkezet*. Hasonló működésű, de a cache adatterületet két egyforma méretű lapra osztják, így két memórialapról tud azonos pozíciójú adatot tárolni. A dolog egyetlen hátránya, hogy a cache-vezérlőnek két directorybejegyzést kell megvizsgálnia, hogy eldöntse, bent van-e az adat. A kétutas cache továbbfejlesztése a négyutas (four-way set), ahol az adatterület már négy részre osztott, vagyis négy memórialapról tárolhat azonos című adatot. Ez már négy vizsgálatot igényel, a vizsgálatok növekvő időigénye miatt nincs még bonyolultabb 8, 16 stb. utas cache. Egyébként, ha a sort így folytatnák, végül a teljesen asszociatív szerkezethez jutnánk.

Egyesített vagy szétválasztott

A processzor kétféle információt hív le a memóriából: adatot és utasítást. Az utasításokat az utasításlehívó tölti be, általában jóval előbb, mint ahogy a végrehajtásban sorra kerülnének. Az adatokat az utasítás-végrehajtó kéri az éppen végrehajtás alatt lévő utasításnak megfelelően. Ezért, ha a memóriához fordulásnál a kettő ütközik, az adatkérés kap elsőséget, mert az adat hiányában az utasítás nem fejezhető be, és ez megakasztja a CPU-t.

Az egyszerűbb, egyesített cache nem tesz különbséget az adat és az utasítás között, mindkettőt ugyanolyan módon tárolja. Azokban az esetekben, amikor a CPU viszonylag kevés utasítással nagy tömegű adatot mozgat — például nagyfelbontású grafikus rendszerek —, az utasítások a cache-ben elveszik a helyet az adatok elől. Ilyenkor — még ha ez bonyolítja is a szerkezetet — célszerűbb külön adat- és utasítás-cache-t létrehozni.

A kétutas (two-way set) cache elve

3. ábra

Cache-méret	Találati arány
1 KB	41%
8 KB	73%
16 KB	81%
32 KB	86%
64 KB	88%
128 KB	89%

Az Intel 486DX és DX2 CPU-i 8 Kbájtos, egyesített, négyutas átíró cache-t tartalmaznak, egy cache-sor 16 bájt. Ugyanilyen a DX4 CPU-ban alkalmazott cache, csak 16 Kbájtos adatterülettel. A Pentium processzorban szétválasztott 8 Kbájtos adat- és ugyanekkora utasítás-cache van, mindkettő kétutas, visszaíró, és egy sor 32 bájt.

Hatékonyság és méret

A cache hatékonyságát, vagyis a hatásfokát azzal mérhetjük, hogy a memóriához fordulások hány százalékában gyorsítja a működést.

Képletszerűen:

Cache-hatékonyság = (cache-találatok/összes memóriához fordulás)*100%

Nem meglepő, hogy mivel az eddig tárgyalt szempontoknak előnyös és hátrányos oldaluk egyaránt van, a hatékonyság döntően a cache méretén múlik. (Lásd a táblázatot.)

A cache meglehetősen költséges, nem mindegy tehát, hogy a memória növelése milyen mértékben növeli a hatékonyságot. Minden program szerkezete más, és az egy lépésben kezelt adatok mérete is változatos, ezért erre nincs számítási eljárás, csak mérési eredmények átlaga. Ezt az Intel-kompatibilis CPU-kra mutatja a táblázat, ami szerint egy bizonyos méret felett a további növekedés már alig jár hatékonyságjavulással.

A mai processzorok mindegyike tartalmaz belső cache-t, amit elsődleges (L1 — level 1) cache-nek neveznek, az alaplapon lévőt pedig másodlagosnak (L2). A másodlagos cache mérete általában többszöröse az elsődlegesének, szerkezetük egymástól független. Például, ha önmagában az elsődlegesnek 70, a másodlagosnak pedig 80%-os találati aránya van, akkor a kettő együtt 94%-os hatékonyságú.

Csórián Sándor

A lapka harmadik oldala

Idegborzoló RAM-regény

Engem is utolért a végzet. Úgy éreztem, nem bírom már tovább az ócska 486/100-as gépemmel, haladnom kell a korral: irány a Pentium! Az elhatározás egybeesett a Compfairrel: ott majd körülnézek, széles a választék, együtt látom a kínálatot. Persze a szaksajtót is elkezdtem ilyen szemmel olvasni. Hol lehet egy kis pluszpénzért egy kis pluszteljesítményt nyújtó összeállítást szerezni, melyek a fontos szempontok. Milyen legyen az alaplap, milyen gyors legyen a processzor, és egyáltalán, mikor lesz már végre — nekem! — Pentiumom.

Miután megszületett bennem a pentiumos elhatározás, csak lassanként alakult ki, hogy mit is akarok. A 100-as Pentium csak egy kicsit drágább a 90-esnél, de gyorsabb, viszont a 120-as és 133-as (főleg ez utóbbi) még primőr, aránytalanul sokba kerülne. Az alaplapon csakis az Intel Triton lapkakészlete lehet, mert az az igazi. És főleg szinkron cache, amelyet pipeline-ként is hirdetnek. Pontos értelmét ma sem tudom, ehhez már nagyon hardveres ész kell, de azt láttam, hogy minden teszt szerint ez a leggyorsabb.

Persze nem veszek kész gépet, hiszen a régivel akkor mit csinálnék, csak veszteséggel lehetne eladni; alaplapcsere, azt én is meg tudom csinálni. Szerencsére még nyáron, jó időben elcseréltem a 4 darab 4 megás 9 bites RAM-omat 2 db 8 megás 32 bitesre. A pentiumos alaplapokba eleve csak ilyen a jó, de 486-osból is elvétve van már csak olyan, amelybe lehet rövid 9 biteseket rakni. A cserébe kapott modulok remekek, 60 nanoszekundumosak, milyen jó lesz az a Pentiumhoz majd — gondoltam akkor.

Persze okos ember nem ugyanott vesz processzort, mint alaplapot, mert így megspórolhat jó pár ezer forintot. Eközben elmagyaráztam egy kollégámnak, hogy okos ember egy helyen vesz meg mindent a gépéhez, sőt: kész konfigurációt vesz, mert ha baj van, akkor tudja, hogy hova szaladgáljon. Ellenkező esetben minden kereskedő — érthető módon — a másikra fog mutogatni.

De velem nem történhet baj, mert én mindent tudok.

Tehát megvettem F-nél a processzort. Lázas telefonok következtek: a kinézett DTK alaplap, szinkron cache-sel éppen sehol sincs, de van egy másik. Nagyon jó referenciák, elfogadható ár, irány a H! Megveszem, az ottani szakember szívességből még a nem nála vásárolt procit is belerakja, mert rendes. Otthon remegő szív, de nem remegő csavarhúzó, mindjárt itt a Pentium. Elsőre apróbb gondok voltak, mert a szakember véletlenül rosszul tette be a processzort, de ezt a hibát korrigáljuk. Hurrá, a gép megy. Boldog vagyok. Előttem egy háromnapos ünnep, lesz módom élvezkedni.

Egyebek közt elindítom a Windowst. Egy-két percig fut, azután elszáll. Ehhez hozzá vagyok szokva, újraindítom. Megint elszáll, de most másik hibával. Ez így megy egész este. A hibák között feltűnően gyakori az „Invalid truetype font", egyébként általam korábban sosem látott hibaüzenet. Az ember megadó, hát installáljuk újra a Windowst, végül is egy alaplapcsere nem kis dolog. Félóra múlva kész, elindítok egy setupot valamelyik programra. Elszáll. Ekkor már kicsit ideges vagyok. Gyors telefon Cs-nek, aki mindent tud. Formázzam újra a winchestert, hiszen új kontroller van benne, az lehet a baj. Újraformázom, újra felrakom a Windowst.

Az a setup végigmegy, de a program, amit fel akartam tenni, azonnal elszáll. Ezt még ez a program eddig sose tette. Elővészek egy tesztprogramot. Szerinte a gépem tökéletes. Megnyugodva elindítom a Windowst, megpróbálok az Interneten körülnézni. A böngésző elszáll. Ezúttal „Divide by zero" a hiba. Újraindítom a böngészőt, hurrá, működik. Csak a Winsock száll el, GPF. Kikapcsolás, másik tesztprogram. A gép jó. De ez a Windows az istennek sem jó.

Az ünnep második napján már kicsit babonás vagyok. Mindent újrainstallálok, semmi extra nincs sem az autoexecben, sem a configban. A hibák záporoznak. Közben meg a DOS-os programok szépen futnak, legalábbis látszólag.

Egy régi emlék miatt a videokártyára gyanakszom. Szaladok, és vasárnap ide vagy oda, sikerül kölcsönkapnom egy másik videokártyát. Szerencsére öt perc alatt bejön az ismerős hibaözön. Kártya mehet vissza a tulajdonoshoz. Még a winchesterrel is lehet baj, meg a modemmel is. A winchestert nem tudom kicserélni, de a modemet kiveszem. Hurrá, nem az a rossz, mert a hibák újra ott vannak. Még megpróbálok sok mindent.

Kikapcsolom a másodlagos cache-t, ettől lelassul a gép, a hiba nem 2 perc alatt jelentkezik, kell neki 3 is. Kikapcsolom a belső cache-t, semmi változás. Lelassítom a memória-hozzáférést, semmi változás.

Kedden reggel első dolgom, hogy felhívjam a H-t, ahol az alaplapot vettem, hiszen csak az lehet hibás. Nem kedvesek, de nagyon készségesek. Adnak egy másik alaplapot. Szerintük egyébként ez egy kitűnő darab, csak jó visszajelzéseik voltak. A szakember azonban, amíg ott vagyok a csere miatt, megjegyzi, hogy ebbe „kétoldalas" RAM kell. Ilyen hülyeséget se hallottam még, gondolom, a RAM az RAM. Hazaviszem az új alaplapot, szerelés, próba, gyönyörűen működik a DOS-ban, de Windowsban jönnek a hibák, nem lehet dolgozni. Már szinte csalódnék, ha nem így lenne. Újabb telefon, ezúttal Á-nak, aki gyakorlott hardveres. Hallott-e ilyen kétoldalas dologról. Soha még ilyet nem hallott, szerinte is hülyeség. Éppen most szereltek Triton chipsetes alaplapokat, gyönyörűen működnek, pedig egyoldalas RAM van bennük.

Próba szerencse alapon délután beviszem a gépet a H-ba. Ott megmutatom az elszállást, eddig is elhitték, de így látványosabb a dolog. Kétoldalas RAM-juk éppen nincs, de betesznek EDO RAM-ot, azzal nem jön be a hiba. Visszatesszük a régit, azzal bejön. Kis financiális tárgyalás kezdődik. A kifogástalan, ámde egyoldalas moduljaimat féláron vennék vissza, az üzlet nekem

kb. 50 ezer forint veszteség. Ez már elég fájdalmas, megtörten távozom. Otthon, „amíg jobb lehetőséget nem találok" felkiáltással visszarakom a régi 486-os alaplapot, az legalább megy. Nos, jön a meglepetés, az sem megy. Bevillan egy régi kép arról, hogy a kártyák lábát időnként meg kell mosni, nem a higiénia, hanem a megfelelő kontaktus miatt. Kis dörzsölgetés, a 486-os megy, mint az álom. Mégis hülyeség az a kétoldalas RAM, hurrá, vissza a Pentiumot. Nem megy. Csak a hibák virulnak.

Újabb szétszedés, újabb dörzsölés, nagyon alapos. A Pentium röhögve produkálja az elszállásokat. Eddig a hibák miatt nem tudtam az Internetről és a Compuserve-ről információkhoz jutni, de most az újra visszatett, kifogástalan 486-ost erre a célra használom. A Compuserve hardverfórumára megeresztek egy levelet a rövid rémtörténettel. Másnapra jön egy válasz, miszerint hülyeség a kétoldalas, neki, a küldőnek nem EDO RAM van a tritonos gépében, remekül megy. Válaszomban megkérem, hogy nézze meg, hány oldalon vannak a chipek. Másnapra megnézi, visszaír, hogy jé, kettő. Ezzel egy időben két újabb választ kapok. Mindkettő megerősíti, hogy nekik sem működött a gép, kicserélték a RAM-ot, azóta rendben van.

Na jó, ez végre már a célegyenes. Beszélek Cs-vel, ők 10% veszteséggel beveszik a régi RAM-ot, rohanok kétoldalast venni. Őrült nagy szerencsém van, még éppen nyitva a bolt, és éppen maradt még nekik két darab 8 MB-os kétoldalas chip. Rohanok velük haza, berakom, izgatottan indítom a Windowst. Megy. És azóta is csak megy, megy. Minden a legnagyobb rendben.

Hogy a rejtélynek mi a fizikai magyarázata, nem tudom. Hogy egy alaplap dokumentációja miért nem említ meg egy ilyen alapvető korlátozást, nem tudom. Hogy az én 10 napnyi szaladgálásomat, fáradságomat, kétségbeesésemet kinek a számlájára írjam, nem tudom.

Epilógus. Kollégám, akit a kész, összeszerelt gépre beszéltem rá, történetemtől végképp meggyőzve vett egy komplett gépet. Mit tesz isten, nem működött. Mármint DOS-ban igen, csak Windowsban szállt el. Pedig nem is Triton lapkakészletes volt a gép. Visszavitte. Ja, a winchester, mondták neki, és kedvesen kicserélték. A gép nem működött. Újra visszavitte, akkor már én is mentem. Kicserélték a memóriát, azóta az a gép is tökéletes.

Horlai János

Újabb keletű fogalmak a DRAM világából

EDO és társai

Ön még nem gondolkodott el azon, hogyan képes egy gyors CPU egy viszonylag lassú DRAM-mal együttműködni? Az elmúlt pár év alatt a CPU-k többször kicserélődtek, míg a DRAM-okban beállt legnagyobb változás az volt, hogy az egyedi chipekről áttértünk a modulokra. A különböző local-bus technológiák többszörösére növelték a CPU kihasználhatóságát, ám a memóriák sebessége alig változott.

Ha belépünk a Pentium-világba, és hagyjuk, hogy a CPU közvetlenül címezze a 60 vagy 70 ns-os DRAM-ot, egy csomó időt várakozásra pazarolunk. A klasszikus megoldás a 15 vagy 20 ns-os statikus cache memóriák alkalmazása. Egy 66 MHz-es Pentiumnak kb. 15,2 ns-ra van szüksége egy memóriaciklushoz: ami viszont túl közel esik a fizikai határhoz. Néhány gyártó különböző típusú SRAM-ot használ.

Nem tart örökké!

Hagyományosan a SRAM-ot aszinkron módon használják, az írás-olvasás sorban egymás után történik, az egyik művelet véget ér, mielőtt a másik elkezdődne. A gyártók lassan azonban átállnak a szinkron, pipeline üzemű SRAM-okra, amelyeknél a második adat előhívása megkezdődik, mielőtt az első ciklus befejeződne. Ilyen szervezéssel elérhető a 100 MB/s-os átviteli sebesség. A szinkron burst SRAM-okkal a ciklusidő durván 10 ns, a 66 MHz-es buszhoz könnyen illeszthető. A 256 K-s külső cache nem tart örökké, szükség van a fő memóriára is.

A fő memória sebességének növelésére két gyógymód kínálkozik: az Enhanced DRAM (EDRAM) és a Cached DRAM (CDRAM). Mindkét hibrid megoldás kisméretű, 15 ns-os SRAM-igényével tűnik ki, s azokat a gyártók minden DRAM-lapkára integrálják. A CDRAM „set-associative cache design" technikát használ, ami kb. 15-20%-kal gyorsabb, mint az EDRAM közvetlen beágyazású megoldása.

Másrészről az EDRAM DRAM része kb. 35 ns-mal fut, durván kétszer gyorsabban, mint a CDRAM-é. Elvileg mind a CDRAM, mind az EDRAM a szinkron burst SRAM-mal összekötve nyújt elsőrendű megoldást. Eddig azonban az alaplapgyártók a CDRAM-ot és az EDRAM-ot a külső cache kiküszöbölésére akarták használni; alternatívaként a hagyományos SRAM/DRAM párosítás mellé.

Nem kell frissíteni

További opció az EDO DRAM. A memóriaműveletek sebességét 30-50%-kal gyorsítja. Meghosszabbítja azt az időt, amely alatt az adat kiolvasható a memóriából, mivel az olvasási parancs nem veszti érvényét, míg egy másik jelet nem kap a chip, vagyis nem kell frissíteni, mint a hagyományos DRAM-okat.

Az EDO DRAM más vezérlést igényel, mint a többi DRAM-megoldás, ezt a vezérlést az Intel Triton chip setje támogatja. Ez ugyan eléggé rendkívüli, de ha egyszer már implementálták, a

MEMÓRIAKEZELÉS



szinkron burst SRAM és az EDO DRAM kombinációja észrevehetően gyorsabb eredményt ad, mint a hagyományos SRAM/DRAM megoldások.

Vannak egyéb lehetőségek is: a Rambus Inc. például az RDRAM-ot ajánlja, amely saját RAM-buszt használ a szokásos memóriabusz helyett. Ez már elég gyors, hogy a Pentium igényeit kielégítse, ám jelentős tervezői munkát igényel az alaplapgyártóktól. Ma az EDO opció van előtérben, de a RAM-igény növekedése holnap akár egy teljesen új technikát eredményezhet.

Az EDO technikát eredetileg a Micron Technology fejlesztette ki. Először is tudnunk kell, hogy a memóriachipekből felépített rendszerekben tömegével vannak egyedileg szelektálandó memóriachipek. Minden egyes chipben minden egyes szó vagy bájt kiolvasható vagy letárolható egy specifikusan címezhető helyen. A mozgatott bitek száma adja a memória „szélességét". A memóriachipek széles választékban érhetők el 1 bittől 64 bit szélességig.

Némi analógiával a címezhető helyek felfoghatók egy postai címzésnek is, ahol külön azonosítjuk az utcát (sorcím) és a házszámot (oszlopcím). Működés közben a számítógépes rendszer mind a sor-, mind az oszlopcímet előállítja a memóriahely azonosításához.

Számos memóriatípus esetében a sor- és oszlopcím más vezérlőbitekkel együtt érvényes. Az EDO DRAM esetében a sorcím (RAS) és az oszlopcím (CAS) arra szolgál, hogy ütemezze a sor- és oszlopcímeket az eszközben. Amikor a cím változik, előáll egy csekély késleltetési idő, ez az „előtöltési" idő. Ami arra kell az elektronikának, hogy végigpásztázza a címzést, feltöltsön egy sor eszközt, amelyek végül is eljuttatják az újonnan címzett hely adatát a chip lábaihoz.

Page Mode

Közös tulajdonságuk a nagy memóriáknak, hogy „postás" technikát alkalmaznak a késleltetés csökkentésére. A page vagy lap az egy sorban levő memóriahelyek összességét jelenti. A page módban csak az oszlopcím változik az egymás után következő memóriahelyek címzéséhez. Ez ismét a postás analógiához vezet: akik házról házra járnak, ahelyett, hogy véletlenszerűen vinnék a leveleket a címzettekhez. A Page módban időt takarítunk meg, mivel a következő memóriahely címzéséhez nem kell változtatni a sorcímet. Az EDO betűszó az Extended Data Output rövidítése, és azt jelenti, hogy az adat hosszabb ideig marad a chip lábain. Ezt egy D flip-flop integrálásával valósították meg, míg a DRAM a mélyben teljesen szokványos. Az EDO DRAM lábain ezt „DQ" jelzés mutatja.

A „Dn" jelzés szokásos az adatkivezetéseknél, míg a „Q" a flip-flopot jelöli. Az EDO csökkenti a memóriahozzáférési időt, mert „latch"-eli és tartja az adatot a „DQ" lábon, miközben a chipben a következő adat felszínre hozása folyik. Ez megengedi egy memóriahely olvasási ciklusának átlapolását a soron következő hely előtöltési idejével. Világos, hogy ilyen technika jótékony hatással van a DRAM sebességére. A Page Mode technika kombinálása az EDO-val megduplázhatja, sőt megnégyszerezheti a sebességet. Az EDO DRAM-ok használata az alaplapgyártóktól új tervezést igényel. Az EDO DRAM-okat sok gyártó kezdi gyártani, és várhatóan sok termékben fognak megjelenni.

Morva Sándor

Közeledés az élő memóriához

„Ehető" számítógép-alkatrészek?

Manapság sok a próbálkozás a számítástechnika területén az egyre változatosabb elektronikai megoldások, hardverkonfigurációk, a feladatokhoz jobban illeszkedő számítógép-architektúrák kifejlesztésére. Egyes fehérjék azon sajátossága, hogy fény hatására bizonyos tulajdonságaik megváltoznak, úgy tűnik, egyszerűbbé teheti az ilyen új architektúrák megvalósításához szükséges hardvert. A számítógép-processzorokban és -tárakban való alkalmazások kutatása a bakteriorodopszin ún. fotociklusára irányul, amely — a fényhatásra adott válaszként — a molekula szerkezetében végbemenő változások sorát jelenti.

Párhuzamos feldolgozási architektúrák létrehozása is szerepel a „legnagyratörőbb" kísérletek céljai között; ezek által több adatcsoport párhuzamos, egyidejű (szinkron) kezelése valósulhat meg. A tárkapacitás növelésére törekedvén olyan hardvert is kifejlesztettek már, amelyik az adatokat a hagyományos kétdimenziós helyett háromdimenziós tárban tárolja. A kutatók neurális hálózatok létrehozásával is foglalkoznak, ezek az emberi agy asszociációs tanulási képességét utánozzák. A mesterséges intelligencia kimunkálása terén van főként jelentőségük az utóbbiaknak.

Egy bakteriális „származású" fehérje

Fehérjealapú vagy fehérjeelemeket is tartalmazó alkatrészek ugyan még nincsenek kereskedelmi forgalomban, de a nemzetközi kutatások sikerei nyomán várhatóan ezek is hamarosan meg fognak jelenni a piacon. Már most előre látható, hogy az a hibrid technológia, amely a félvezető lapkákat és a biológiai eredetű fehérjemolekulákat kombinálja, a fantázia világából hamarosan a kereskedelmi-műszaki felhasználás valóságába kerül. A folyadékkristályos képernyők technológiája révén napjainkban is találhatunk példát a hasonló — már megvalósított és kereskedelmi, pénzügyi sikereket eredményező — hibrid rendszerekre. A legtöbb laptop olyan folyadékkristályos képernyőt alkalmaz, amely a fényerő szabályozásához mind félvezető lapkákat, mind pedig szerves molekulákat felhasznál.

Jelenleg többféle biológiai eredetű molekula felhasználhatóságát is vizsgálják. Ezek közül pillanatnyilag a bakteriorodopszin kecsegtet a legtöbb sikerrel. A bakteriorodopszin felé először a hetvenes években fordult a kutatók figyelme, amikor kiderült, hogy fényhatásnak kitéve egyes fehérjék különleges, megváltozott tulajdonságokat mutatnak.

Először szovjet tudósok vették észre és dolgozták ki a bakteriorodopszin számítástechnikai alkalmazásának a módját. A biomolekuláris elektronika kifejlesztésén dolgozó csapatnak sikerült meggyőznie az akkori szovjet katonai vezetést, és elnyerték a támogatásukat a bioelektronikai kutatásokhoz. Az elindított „Rodopszin-projekt" részleteit még mindig homály fedi, azt még ma is katonai titokként kezelik. Ma csak annyit lehet tudni, hogy mikrofilmtárat és processzorokat fejlesztettek ki a bakteriorodopszin-kísérletek során.

Amerikában is a hetvenes években indultak a bakteriorodopszinnal kapcsolatos kutatások.

A nyugalmi és a gerjesztett...

A nyugalmi állapotban lévő molekulát, illetve állapotát bR-rel, a gerjesztett állapotokban levő (a nyugalmi állapotból kimozdított és ismételten nyugalomba még nem került, általában instabil állapotú) molekulákat az ábécé különböző betűivel jelölik. A számítógépes alkalmazásokra irányuló kísérletek zöme a molekula egyes különböző tulajdonságú állapotait próbálja meg adatbitek megjelenítésére felhasználni. Egyes állapotokban a molekulák bizonyos — az állapotra jellemző — hullámhosszúságú (színű) fénysugarakat elnyelnek. Következésképp: adott hullámhosszúságú lézersugárral megvilágított molekulákat vizsgálva, hogy melyik molekula nyeli el, és melyik engedi át a fénysugarakat, visszaolvashatjuk az állapotokat. Vagyis azt, hogy melyik molekula tartalmaz ehhez az adott hullámhosszhoz, tehát az ahhoz tartozó állapothoz is rendelt bitértéket.

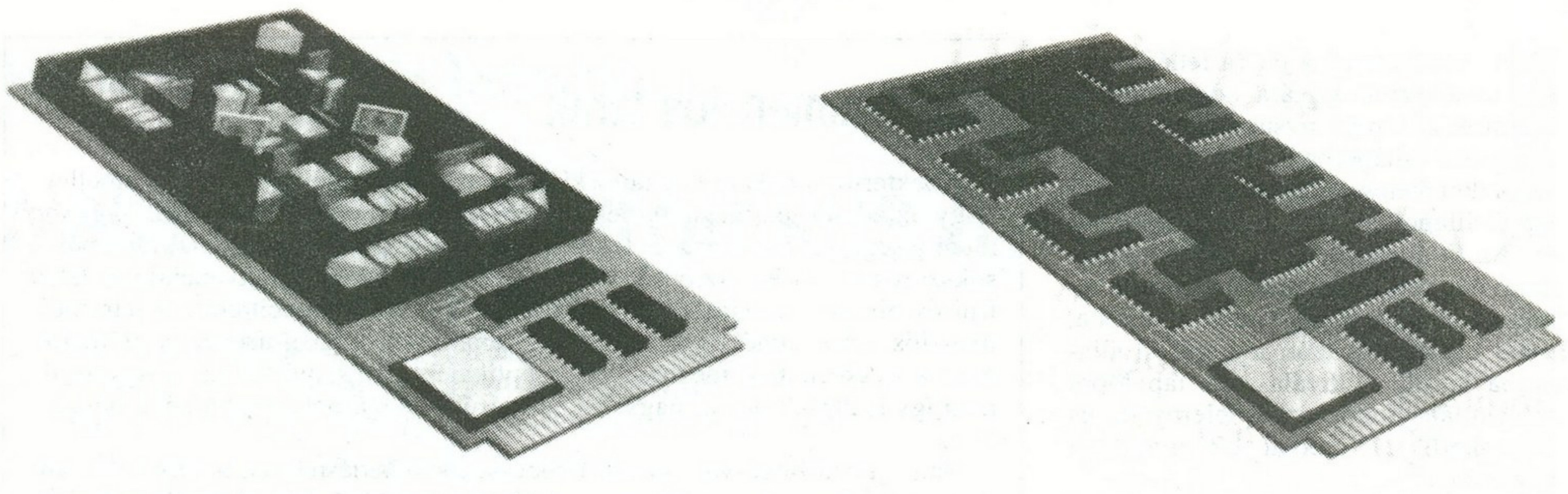
A kutatások során általában a bakteriorodopszin nyugalmi és valamelyik gerjesztett állapotát rendelik a 0-s és az 1-es értékhez. Az állapotok közötti váltást lézersugárral vezérlik.

Az első próbálkozások a bR és K állapotokkal foglalkoztak, de ezek nagy hátránya volt, hogy csak nagyon alacsony hőmérsékleten, folyékony nitrogénnel hűtve tudták megbízhatóan vezérelni a fénysugár indukálta állapotváltásokat. E kezdeti kísérletek során kialakított készülékek sokkal gyorsabbaknak bizonyultak, mint a hasonló hagyományos, félvezető alapú kapcsoló áramkörök. A bR-K állapotváltás 10^{-12} másodperc, míg a félvezető áramkörök kapcsolási ideje 10^{-9} másodperc nagyságrendű.

Állapotok — szobahőmérsékleten

A bakteriorodopszinnal folyó újabb kísérletek már olyan gerjesztett állapotokkal folynak, amelyek szobahőmérsékleten is működő és használható eredményt produkáltak.

Fény hatására a bakteriorodopszin gerjesztett állapotba hozható, majd újabb megvilágítás hatására ismét fényt (energiát) nyelve el, a szerkezete ismét megváltozik, és egy újabb gerjesztett állapotba kerül. Ezt a folyamatot fotonspecifikus szekvenciális folyamatnak nevezzük. Ilyen reakció játszódik le például az O jelű gerjesztett állapotból a P, majd a Q állapotokba történő átmenet során. Ezek a szerkezetek, ill. állapotok két egymást követő — előbb



Elképzelhető, hogy a jövő számítógépeiben szerves–szervetlen anyagú hibridkártyák lesznek, és a félvezetőket (világos részek) proteinek veszik majd körül (sötét részek), háromdimenziós memóriablokkot alkotva. (Illusztráció a Scientific Americanből.)

zöld, majd vörös — fényimpulzus hatására alakulnak ki. A bR nyugalmi állapotú molekulák elnyelik a zöld fénysugarakat, és a megvilágítás hatására K gerjesztett állapotba kerülnek, amelyből nagyon gyorsan M, majd O állapotba ernyednek el. Az O gerjesztett állapotú molekulák rövid idő múltán bR nyugalmi állapotba ernyednek, hacsak nem éri őket egy újabb, vörös fényhatás. Az O állapotú molekulák elnyelik a vörös fényt, és hatására P gerjesztett állapotba kerülnek. A P állapot rövid élettartamú, gyors elernyedés után Q állapottá változik, ami már stabilnak tekinthető, akár több évig is változatlan maradhat.

Itt tehát egy elágazó, fotonspecifikus szekvenciális folyamattal, illetve reakciósorozattal van dolgunk. A bR és a Q állapotoknak, éppen nagy stabilitásuk miatt van jelentőségük a nagy sűrűségű, tartós tárak, memóriák előállítására irányuló kutatásokban.

A háromdimenziós tár

Úgy tűnik, a fotonspecifikus szekvenciális folyamattal létrejövő P és Q gerjesztett állapotok különösen jól hasznosíthatók majd az adatok párhuzamos (szinkron) feldolgozását végző párhuzamos processzorok egyszerre nagy mennyiségű adatot igénylő működése során. Ilyen megközelítésben egy újfajta tár megjelenéséről is beszélhetünk, a háromdimenziós tárról, amelybe nagy mennyiségű adatot lehet egyszerre, egy ciklusban, párhuzamosan beírni, és belőle kiolvasni.

A gyakorlatban e célból egy bakteriorodopszin-kockát két egymásra merőleges egy-egy lézersugárforrás-hálóval, -ráccsal fognak közre. Az egyik lézerháló a háló minden egyes rácspontján zöld fényt tud kibocsátani. A háló egyik sorát működtetve a kocka egyik (metsző) síkjában levő bakteriorodopszin molekulákat zöld lézersugarakkal világíthatjuk meg. Ez a zöld megvilágítás a kocka kiválasztott, megvilágított síkjába eső fehérjemolekulákban egy fotociklust indít el, a nyugalomban lévő fehérjemolekulák K állapotba gerjedés után M, majd O állapotba ernyednek. A bR, nyugalmi állapotban lévő molekulák elnyelik a zöld fényt, de utána, gerjesztett állapotba kerülve már szabadon átengedik, így a síkba eső, egymást takaró molekulák, egymás után sorra billennek ki nyugalmi állapotukból. Végül (néhány ezredmásodperc múlva) a síkba eső összes (megvilágított) molekula O állapotba kerül.

Az O állapot élettartama a megelőzőeknél valamivel hosszabb, így ez az az időpont, amikor az először megvilágított molekulák még, az utoljára megvilágítottak pedig már O állapotban vannak. Ezzel tehát kiválasztottuk és O állapotba gerjesztettük a kocka egyik síkjában lévő molekulákat, ezt — mint egy könyvnél — a kocka egyik oldalának, lapjának is tekinthetjük, ezért a folyamatot lapkiválasztásnak, a zöld lézer sugárforrásokat lapozó lézereknek is nevezik.

Most kell működnie a másik, vörös fényt kibocsátó lézerhálónak. Ezt a lézerhálót úgy kell vezérelni, hogy csak azokon a rácspontjain bocsásson ki

Bakteriorodopszin

A *rodopszin* egy összetett fehérje, amely az emlősök szemének recehártyájában (a retinában) is megtalálható. A rodopszin és úgyszintén a bakteriorodopszin olyan bonyolult szerkezetű, összetett fehérjék, amelyek egy kromofor nevű komponenst is tartalmaznak. A kromofor elnyeli a fény energiáját, ezzel bonyolult belső változások, mozgások sorozatát váltja ki a molekulában, miáltal az összetett fehérje teljes szerkezetében gyors és lényeges változásokat eredményez. Ezek a változások a fehérje optikai és elektromos tulajdonságait is megváltoztatják. Az emberi szemben például, amikor a rodopszin fényt nyel el, a fehérje szerkezetében bekövetkező változás energiafelszabadulással jár, ami elektromos jelként alkalmas arra, hogy az agy részére vizuális információkat juttasson el.

Csak a későbbiekben terelődött a figyelem a *bakterioropdopszin* felé, mert nagyobb a stabilitása, kedvezőbbek az optikai tulajdonságai, és nem utolsósorban könnyebb előállítani nagy mennyiségben. A számítógépek alkatrészeinek ugyanis el kell tudniuk viselni a környezeti viszonyokban beálló extrém változásokat, mostoha körülményeket is. A bakteriorodopszin a természetben sós mocsarak élővilágában fordul elő, ahol a hőmérséklet a 70 fokot is meghaladhatja, és a molekulák gyakran erős fényhatásnak is ki vannak téve.

A Halobacterium Salinarium sejtmembránjában lévő bakteriorodopszin lehetővé teszi, hogy a baktérium olyan alacsony oxigénkoncentráció esetében is szaporodjon, amely egyébként nem volna elegendő az életben maradásához. Fény hatására a bakteriorodopszin-fehérje (protein) megváltoztatja a szerkezetét, a sejtmembránon keresztüllök egy protont, és ezzel szolgáltatja a sejt anyagcseréjének a fenntartásához szükséges energiát.

fényt, amelyeknek a tár (a fehérjekocka) kiválasztott lapjára adatbitet kell beírniok. A vörös fénysugarakat a kocka nyugalmi állapotban lévő molekulái szabadon átengedik, csak a kiválasztott lap O állapotban lévő molekulái nyelik el. Az O állapotú, vörös lézersugárral megvilágított molekulák szerkezete ismét átalakul a P gerjesztett állapotba, majd elernyedés után a stabil Q állapotba kerül. A kiválasztott lap többi molekulája fokozatosan elernyed, és visszakerül az eredeti bR nyugalmi állapotába.

A P és Q állapotokhoz hozzárendelhetjük a bináris 1-es értéket, a bR és O állapotokhoz pedig a bináris 0-t. Ily módon a kiválasztott lapra nagy mennyiségű adatot — egy teljes lapot — lehet egy ciklusban párhuzamosan beírni.

Háromdimenziós tárak

A bakteriorodopszin-kockából kialakított háromdimenziós tárak — amellett, hogy az adatok párhuzamos feldolgozását megkönnyítik — sokkal nagyobb mennyiségű adat tárolását is lehetővé teszik. Még az optikai (CD) táraknál is sokszorosan többet. Az optikai tárakat csak sávonként, szekvenciálisan lehet írni és olvasni, maximális adatsűrűségük kb. 100 megabit/cm^2. A háromdimenziós tárak elméletileg akár az 1 terabit/cm^3 adatsűrűséget is elérhetik, amit a gyakorlatban mechanikai és optikai korlátok mérsékelnek ugyan, de még így is 200-300-szor nagyobb adatsűrűséget ígérnek, mint a kétdimenziós tárak.

Nagy jelentősége van annak a sebességnövekedésnek is, amely a háromdimenziós táraknál a párhuzamos feldolgozás lehetőségével kínálkozik. Egy 1 Kbit élhosszúságú fehérjekockába egyszerre (egy lapra) 1 Mbit adatot lehet beírni. A teljes írási ciklus, mint korábban láttuk, 10 ms, ami tehát 10 Mbit/s írássebességet jelent. Ha azt is figyelembe vesszük, hogy egy tárba több ilyen kocka is beépíthető, az egyszerre, párhuzamosan működő tárak sebességei pedig összeadódnak, a 10 fehérjekockából kialakított tár sebessége eléri a 100 Mbit/s értéket.

A tárolt adatok kiolvasása

Az adatvisszanyerés az előbbiekben ismertetett tárból azon alapul, hogy az O állapotú fehérjemolekulák elnyelik a vörös fénysugarakat, míg a többi állapotban nem. Az adatok kiolvasása — a beíráshoz hasonlóan — a lap kiválasztásával, azaz zöld fényű megvilágításával kezdődik. Zöld fény hatására a 0-s bitértékű, bR nyugalmi állapotú molekulák O gerjesztett állapotba kerülnek, míg az 1-es bitértékű, P és Q állapotú molekulákra a zöld fénysugár nincs hatással, és természetesen ezek a molekulák nem is nyelik el a zöld sugarakat. Így a kiválasztott lapon a 0-s bitértékű helyeken a molekulák O, az 1-es bitértékű helyeken pedig P és Q állapotban lesznek.

Ha a zöld megvilágítás után kb. 2 ezredmásodperccel a vörös lézerháló minden rácspontjából nagyon gyenge intenzitású vörös fénnyel világítjuk meg a fehérjekockát, a sugarakat a kocka minden molekulája át fogja engedni, kivéve a kiválasztott lap 0-s bitértéket képviselő O állapotú molekuláit.

Az O gerjesztett állapotú molekulák elnyelik a vörös fénysugarakat, de ha a fényerőt és a megvilágítás idejét elég kis értékeken tartjuk, az így befogott energia nem elég ahhoz, hogy a molekulák szerkezete is megváltozzék.A kockának a vörös fény lézerhálóval szembeni oldalán elhelyezett érzékelő hálóval pontról pontra, bitről bitre érzékelhetjük a kockán áthaladó fénysugarakat (1-es bitértékű pontok), és az elnyelt fénysugarakkal szemben az árnyékolt pontokat (0-s bitértékű pontok). A fényérzékelő háló tehát egy lap összes adatát olvassa ki egyszerre, párhuzamosan, egyetlen olvasási ciklusban.

Az olvasási folyamat kb. 10 ezredmásodperc alatt zajlik le, ami kb. 10 Mbit/s-os laponkénti olvasási sebességet jelent.

Az előbbiekben ismertetett módon tehát egy gyors és nagy kapacitású, egyszer írható vagy csak olvasható tárhoz jutottunk. Ilyen tárban akár egy nagy enciklopédia adatai is elférnek, cserélhető változat esetén a kis fehérjekockákat az alkalmazások szerint csak ki kell cserélni, és máris egy másik hatalmas mennyiségű adatot tartalmazó tár információit dolgozhatjuk fel.

Más-más tárak

A bakteriorodopszin kockákból újraírható tárak is kifejleszthetők. A már beírt adatokat felülírni természetesen csak úgy lehet, ha a kérdéses lap adatait először töröljük. A törlés a lap kék fényű megvilágításával történik. A kék színű fénysugarakat a Q gerjesztett állapotú, 1-es bitértéket képviselő molekulák elnyelik, a szerkezetük átalakul, és bR nyugalmi állapotba kerülnek. A fehérje Q-tól különböző állapotaira a kék fénysugarak nincsenek hatással.

Törölhető tár megvalósítása érdekében a lapkiválasztó hálóval szemben kék (törlő) lézersugarakat kibocsátó hálót kell elhelyezni. Ennek szerkezete és működése is hasonló a lapkiválasztó hálóéhoz. A törlési ciklus egy egész lapot töröl, ezért — ha a lapnak csak egy kis részén szeretnénk az adatokat átírni a többi adat változatlanul hagyása mellett — a törlési ciklust egy olvasásnak kell megelőznie, és a kiolvasott adatok módosítása után írási ciklusnak kell követnie.

Űr — és a távlatok

A pontos és biztonságos írás és olvasás érdekében a fehérjekockáknak nagyon homogéneknek kell lenniük, azaz a molekulák eloszlása a kockán belül egyenletes kell, hogy legyen. Ez a homogén szerkezet kis gravitációjú térben zajló előállításukkal érhető el. A NASA és néhány egyetem összefogásával már két űrhajós kísérletet hajtottak végre, amelyek a homogén bakteriorodopszin előállítását is elvégezték. A kísérletek sikeresek voltak, és még továbbiakat is terveznek. A kísérletek sikeres befejezése után várható az űrben előállított bakteriorodopszin ipari célú felhasználása is.

Jelenleg kísérletek folynak a bakteriorodopszinnal az asszociatív tárak irányában is. Az asszociatív tár úgy működik, hogy egy adathalmazt — általában egy képet, a kép pontjainak egészét — tekint egy adatnak, és az egész tárat végigpásztázza, míg csak egy másik azonos adathalmazt nem talál. Sok esetben nem talál teljesen azonos adathalmazt, ilyenkor a legjobban hasonlító adathalmazt keresi meg, és mint egy tanulási folyamatban, azt választja ki egy lehetséges válaszként.

Az asszociatív tárak esetében a bakteriorodopszin-vékonyrétegek holografikus tulajdonságait használják fel. A hologramok lehetővé teszik több kép tárolását is a tárnak ugyanazon a kis részén (mintegy egymás alatt), ezáltal könnyebbé válik azok összehasonlítása.

Dobó Csaba

Atomizmus a digitális technikában

Nem csak elektronikus a világ

A gépi információkezelés (tárolás, feldolgozás, továbbítás stb.) sokféle módon megoldható. Fontos tudni, hogy már a digitális körön belül sem csak elektronikus, hanem más, például mechanikus, elektromechanikus, optikai, hidraulikus és pneumatikus eszközöket is használ az információkezelés-technika, és az effajta eszközök valószínűleg sosem fognak kimenni a divatból, ugyanis sok olyan terület van, ahol előnytelen vagy lehetetlen az elektronikus megoldások alkalmazása.

A nem elektronikus információkezelő rendszerek nem csak műszaki okokból érdemelnek figyelmet. Ezek építőelemeinek működését nemegyszer azonnal át lehet látni, és ezért pl. bizonyos pneumatikus elemekkel sokkal könnyebb a digitális rendszerek működésének megértetése és szemléltetése, mint elektronikus elemekkel, amelyek működése elméleti villamosságtani ismeretek nélkül érthetetlen.

Egy „univerzális” műveleti elem

A következőkben bemutatunk egy pneumatikus digitális műveleti elemet. (Sokféle különböző elven működő pneumatikus digitális műveleti elem van, ez egy a sok közül.) Ez a műveleti elem „univerzális” abban az értelemben, hogy egyedül ennek (és persze összekötő vezetékeknek) a használatával minden más bináris műveleti elem felépíthető. (Nem csak egy „univerzális” elem létezik!)

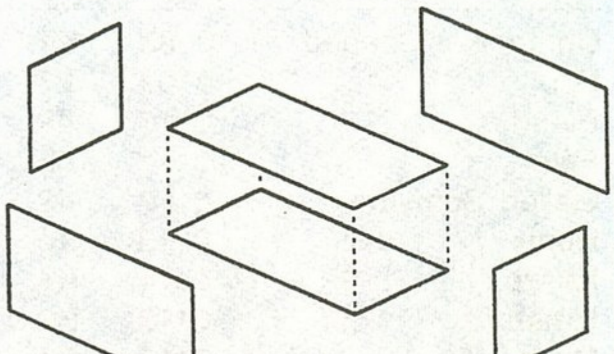
Még érdekesebb az a tény, hogy ha a teljes bináris hardvert egy bonyolult vegyületnek fogjuk fel, akkor ez a „vegyület” felépíthető egyetlenegy fajta „atom” különböző kapcsolásaival is, és erre a célra az általunk bemutatott „univerzális” pneumatikus digitális műveleti elem is alkalmas.

(Rokon jelenség tapasztalható a szoftver és annak részei esetében. Bármely program olyan, hogy valamiktől függően, valamik hatására valamilyen viselkedésre kényszeríti a gépet, sőt minden program maga is felépíthető ilyen tulajdonságú elemekből. A felépítés az ún. „programozás”. A szoftver esetében sokféle „atom” van ugyan, de mindegyiknek azonos a működése: valamiktől függően, valamik hatására valamilyen viselkedésre kényszeríti a gépet.)

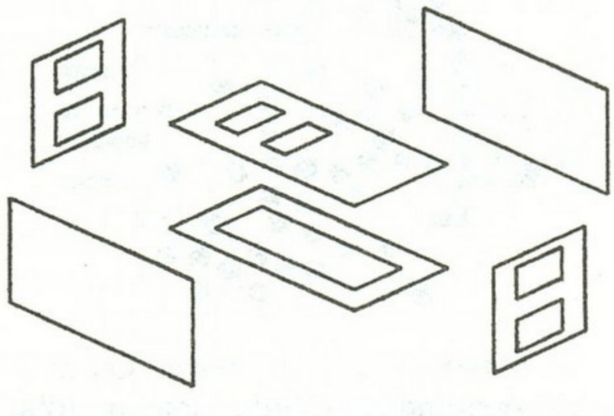
„Univerzális” pneumatikus digitális műveleti elemünk egy NEM VAGY, másszóval NOR „kapu”, amelynek „univerzális” voltát minden tankönyv tárgyalja, ezért ezt itt nem bizonyítjuk. Amivel bővebben foglalkozunk: ennek az elemnek a felhasználása információtárolásra (egy ún. „RS flip-flop” felépítésére).

NEM VAGY elem, és a tár-elem

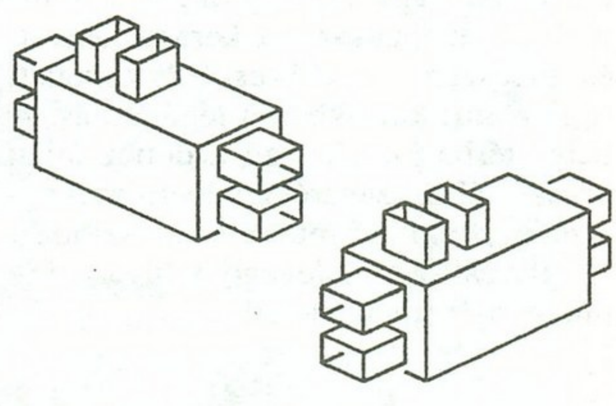
A tárgyalt pneumatikus digitális műveleti elem valóságos, gyakorlatban használt eszköz, a rajzok azonban nem műszaki rajzok, hanem elvi vázlatok, céljuk a működés lényegének a szemléltetése. A mondanivalót a könnyebb érthetőség kedvéért „képregény” formába rendeztük. A rajzok számozva vannak. Az egyes rajzok és a hozzájuk tartozó szöveg:



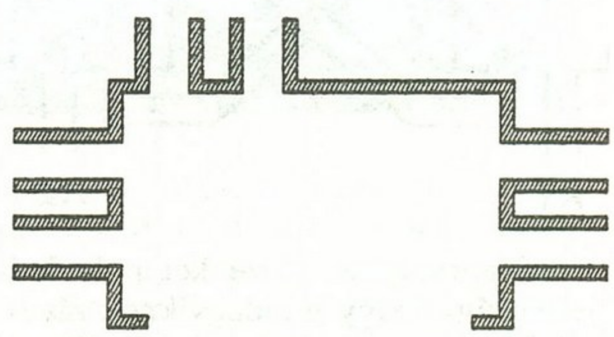
1. A műveleti elem magja egy közönséges doboz, téglalap oldalakkal.



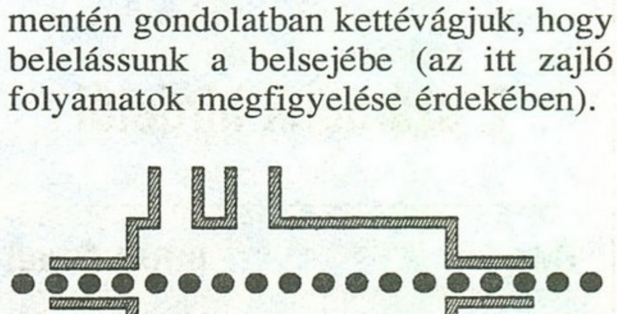
2. A doboz egyes lapjaira (kivéve az oldallapokat) lyukakat vágunk.



3. A lyukakra — az alsót kivéve — kürtőket, kéményeket rögzítünk. Ezekre csatlakoznak majd a vezetékek, amelyek egyszerű csövek, példánk esetében téglalap keresztmetszettel.

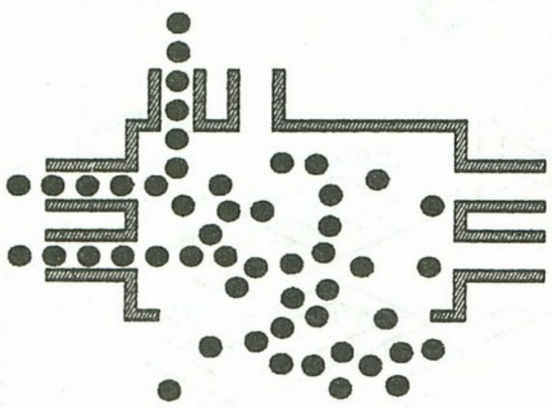


4. A berendezést szimmetriasíkja mentén gondolatban kettévágjuk, hogy belelássunk a belsejébe (az itt zajló folyamatok megfigyelése érdekében).

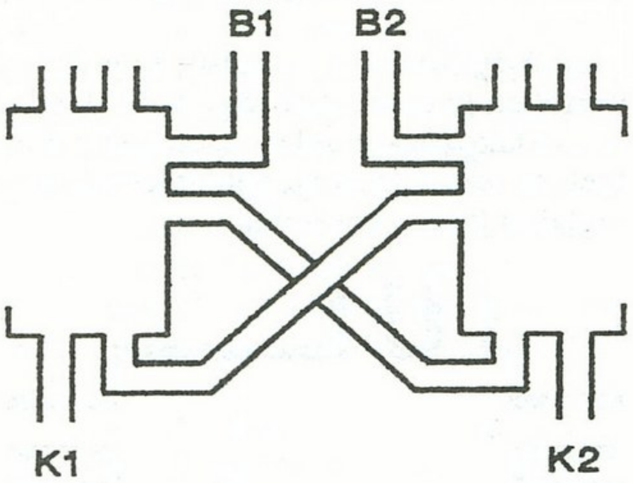


5. A berendezés „nyugalmi állapotában” úgy működik, hogy egymás mellett két egyenletes légsugár áramlik rajta keresztül. Zavartalan „keresztthu-

zat" van benne. (A gázmolekulákat valódi méretükhöz képest óriási gömbök szemléltetik.) A nyílásokon állandóan áramlik a levegő (ez a táplevegő-áramlás kicsit hasonlít az elektromos tápenergiához).



6. A berendezés működése két „vezérlő operandus" (elvileg lehetne több is) révén vezérelhető, a B1 és a B2 kürtőn keresztül. Ha ezek egyikébe, másikába vagy mindegyikébe belefújunk, ez megzavarja a keresztfhuzatot, és megszűnik a K1 és K2 kimeneti (operátum) kürtőkben a légáramlás. A belső térbe jutó levegő a doboz alján távozik. (Ha a zavaró oldalbefújás megszűnik, rövid idő múlva a kereszthuzat is helyreáll.) A táplevegő befúvása egy pillanatig sem szünetel!



7. Kapcsoljunk össze két műveleti elemet úgy, hogy mindegyiknek rávezetjük az egyik kimeneti kürtőjét a másik valamelyik vezérlő kürtőjére! Így egy olyan rendszert kapunk, amelynek 2-2 tápbemenete, 1-1 vezérlőbemenete, és 1-1 vezérelt kimenete van. Ha a B1 bemeneti csőbe belefújunk, akkor ennek hatására a K1 csövön megszűnik a kiáramlás (ha volt), a K2 csövön pedig megindul (ha nem volt), és ez az állapot akkor is fennmarad, ha a B1 csőben már nincs áramlás. Ha viszont a B2 bemeneti csőbe fújunk bele, akkor a K2 csövön szűnik meg a kiáramlás (ha volt), a K1 csövön pedig megindul (ha nem volt), és ez az állapot akkor is fennmarad, ha a B2 csőben már nincs áramlás. A vezérlő bemeneteknek mindegyikébe egyszerre is belefújhatunk. Ha egyszerre fújunk bele mindkettőbe, úgy, hogy egyszerre hagyjuk abba a belefújást, akkor ennek a műveletnek a kimenetekre való hatása kiszámíthatatlan lesz. Ezért ezt a lehetőséget a gyakorlatban sosem használják, illetve használatát tiltják.

Szabályos használat mellett, a kimeneteken tapasztalható áramlásokból meg tudjuk mondani, hogy utoljára melyik vezérlő kürtőben volt áramlás. Tehát ez az elem megőrzi (feljegyzi, regisztrálja) ezt az információt. (Melynek információelméleti mennyisége, „információmennyisége" pontosan 1 bit, ezért az ilyen tárolóelemeket 1 bites regiszternek is hívják.)

Mi van akkor, ha...?

A gondolkodó embert egy kicsit zavarja, ha nem tudja teljesen áttekinteni valaminek a működését. (Például mi is történik akkor, amikor mindkét vezérlő kürtőbe belefújunk?) Ez a kielégítetlenségérzés minden műszaki alkotás esetében megvan. Minden műszaki alkotásnak van ugyanis (rendeltetés szerinti) működési és működtetési tartománya, amelyen kívül működési bizonytalanságok vannak, és károk fenyegetnek. Örülni szoktunk, ha valami valamire használható, és kiaknázzuk ezt a használhatóságot. Itt most mi valamit valamilyen módon valamire sikeresen felhasználtunk. Hogy a használata más módon milyen lehetőségeket rejt, az más kérdés.

A műszaki alkotás nem egy egyszerű matematikai függvény, amelyet pontosan ki lehet elemezni. Már egy egyszerű levesestányér esetében sem tudjuk mindenre kiterjedően, hogy milyen esetben hogyan viselkedik, milyen irányból mekkora igénybevételt bír ki. A lényeg az, hogy a meghatározott célra jól lehessen használni. (Témánknál maradva, ha valakinek kedve van, azzal a kérdéssel is foglalkozhat, hogy mi történik akkor, ha a vezérlő kürtőkbe nem belefúj, hanem megszívja azokat. Lehet, hogy ebből is „kijön" valami használható.)

Félreértés ne essék! Csodálatos dolog lenne, ha minden eszközünkről teljeskörűen tudnánk, hogy mikor hogyan viselkedik. Ez az igény azonban sosem lesz kielégíthető. Fontos igények kielégítésére kell eszközöket keresnünk, amelyeknek másféle körben való viselkedésének kutatásáról (lehet az bármilyen értékes) valamilyen ok miatt (legalábbis időlegesen) mindig le kell mondanunk.

A digitális berendezések regiszterei működésének megértéséhez három dolgot kell érteni. Hogyan történik a bejuttatás? Hogyan történik a tárolás? És hogyan történik a kiolvasás, az „emlékek felidézése"? (Mondjuk például egy regiszter tartalmának egy másik regiszterbe való átmásolása.) Az első két kérdésre megadtuk a választ. A harmadik azonban már nem tárolási, hanem információáramoltatási téma, de ez is egyszerűen megoldható „univerzális" elemünkkel. Érdemes egy kicsit elgondolkodni, hogyan.

Pogány Csaba

Interaktív Video Magazin

A multimédia mint média

Amikor egy számítástechnikai eszközt vagy programot olyan rétegnek szánnak, amelyiknek nem sok köze van a számítástechnikához, az első spontán vélemény az, hogy bátor emberek lehetnek a kibocsátók. De ha szerencséjük van, akár sikeresek is lehetnek. Nos, ilyen bátorságról tett tanúbizonyságot az Irisz Könyv- és Lapkiadó Multimédia Stúdiója, amely elkészítette és útjára bocsátotta (a mai világban kissé húzós, 1990 Ft-os áron) az Interaktív Video Magazin első számát — Falus Tamás szerkesztésében.

A lap megtekintéséhez Windows 3.11 vagy Windows 95 szükséges, valamint Video for Windows, amelyet — ha nincs — a lemezről is rögtön telepíthetünk. A kiadvány első próbálkozás. Ez tükröződik abban is, hogy a filmhatás kiváltására szinte teljesen alkalmatlan Video for Windows technikát választották a videoanyag megjelenítésére. A gyufásdoboznyi kép ui. csak a mozgás képzetét keltheti. Mielőtt a rendszert megtervezték, érdemes lett volna tájékozódni, az ennél jóval látványosabb Intel Indeo vagy Qpeg/Jpeg Quicktime-alapú videorendszerek felhasználásáról.

A CD szerzői e zavaró hibától eltekintve mindent megtettek annak érdekében, hogy legyen kedvünk órákat eltölteni a folyóirat böngészésével. A fő témaköröket a virtuális videokazettának a virtuális videolejátszóba helyezése után lapozhatjuk fel. Az adott témakörökön belül azután ott van számunkra a videó, a kép és a hipertextes szöveg vallatásának lehetősége.

Van benne katalógus, van egy igen hosszú interjú George Lucasszal a Csillagok háborújának rendező-trükkmágusával... Mind olyan téma, amely helyet kaphat egy multimédiában. Miként a filmismertetések, előzetesek, tartalmi kivonatok is. Ezek nagy kincsnek számítanak a filmrajongók számára.





A multimédia a Video Magazin papírfolyóirat bázisán mintegy annak folytatása, kiterjesztése. Így ilyen lapokhoz hasonlóan megtalálhatjuk benne a fontosabb cikkeket, és a korábbi számok címlapját. A kiadvány a számítástechnika kedvelői mellett a videofilmek és az „igazi" filmek kedvelőinek is sok élményt nyújthat, adatbázisa viszont az évi magyarországi mozi- és videoforgalmazás alapvető forrása. Így mindenkinek érdemes megszereznie, aki a téma iránt érdeklődik, és van legalább „középerős" multimédiás, windowsos gépe.

Kis János

A mi lemezünkön is — m. v.

Floppy Információs Lap

Ez a kiadvány ugyancsak egyike a „renitens" sajtótermékeknek. Adathordozó-választása és neve az egykor volt FLOPPY.LAP-ra asszociál, de tartalmilag inkább tekinthető az ugyancsak a Cédrus Rt által 4-5 évvel ezelőtt kiadott másik két floppyújság, a BANKÁR és az IRÁNYTŰ utódjának. A Floppy Információs Lap feladata is az, hogy információival hatékonyabbá tegye a vállalkozások munkáját. (Egy korlátozott terjedelmű bemutatkozó változatot mostani számunk lemezmellékletére is elhelyeztünk.)

A Floppy Információs Lap ötvözni igyekszik a hagyományos újságok és a számítógép által kínált előnyöket. Csak számítógépen olvasható formában jelenik meg, az előfizetők igényeihez alkalmazkodva heti, illetve havi frissítéses verzióban.

Általa az előfizetők mindenféle hálózattól függetlenül juthatnak hozzá az őket érdeklő gazdasági információkhoz. (Ha belegondolunk, a magyar gazdasági élet tempója általában nem is feltétlenül igényli a napi információfrissítést, legfeljebb bizonyos részterületeken nélkülözhetetlen a piaci információk online elérése.)

Profilját tekintve a lap kifejezetten gazdasági-vállalkozási orientációjú, és jelenleg mintegy 30 rovata van. A hasonló témákra szakosodott nyomtatott lapoktól eltérően publicisztikai jellegű írásokat, cikkeket nem közölnek, csak „száraz" adatokat, tényeket, gazdasági információkat. Elsődleges szempontjuk az információk aktualitása és megbízhatósága.

A Floppy Információs Lap egyik legfontosabb célja, hogy naprakész (pontosabban: „hétrekész") tanácsokkal szolgáljon az érvényben lévő adó-, tb-, vám-, munkavédelmi jogszabályokkal, rendelkezésekkel kapcsolatban. Ezen túlmenően tartalmazza a fent említett jogszabályok egységes rendszerbe foglalt, gyakorlatban alkalmazható, közérthető magyarázatát is.

Néhány további témakör a floppy-lap kínálatából:

— Európai és ázsiai vállalatok gazdasági ajánlatai a külpiacok felé akár értékesítési, akár beszerzési céllal orientálódó magyar vállalkozások számára.

— Külföldi társadalombiztosítással, adózással, cégalapítással, munkavállalással kapcsolatos jogszabályok főleg a külföldön vállalatot létrehozni szándékozó cégeknek.

— A Világbank, illetve a Phare program által finanszírozott tenderekkel kapcsolatos tájékoztatás, versenytárgyalási felhívások, részvételi feltételek.

— A privatizációs kínálattal és a normál ingatlanpiaccal kapcsolatos adatállományok, elsősorban beruházási vállalkozások, új telepet létesíteni szándékozó cégek figyelmébe.

— Aktuális lista a felszámolás, végelszámolás, csődeljárás alatt álló cégekről.

— Magyarországon bejegyzett védjegyek.

— Magyarországon bejegyzett találmányok.

— A Gazdasági Versenyhivatal döntései.

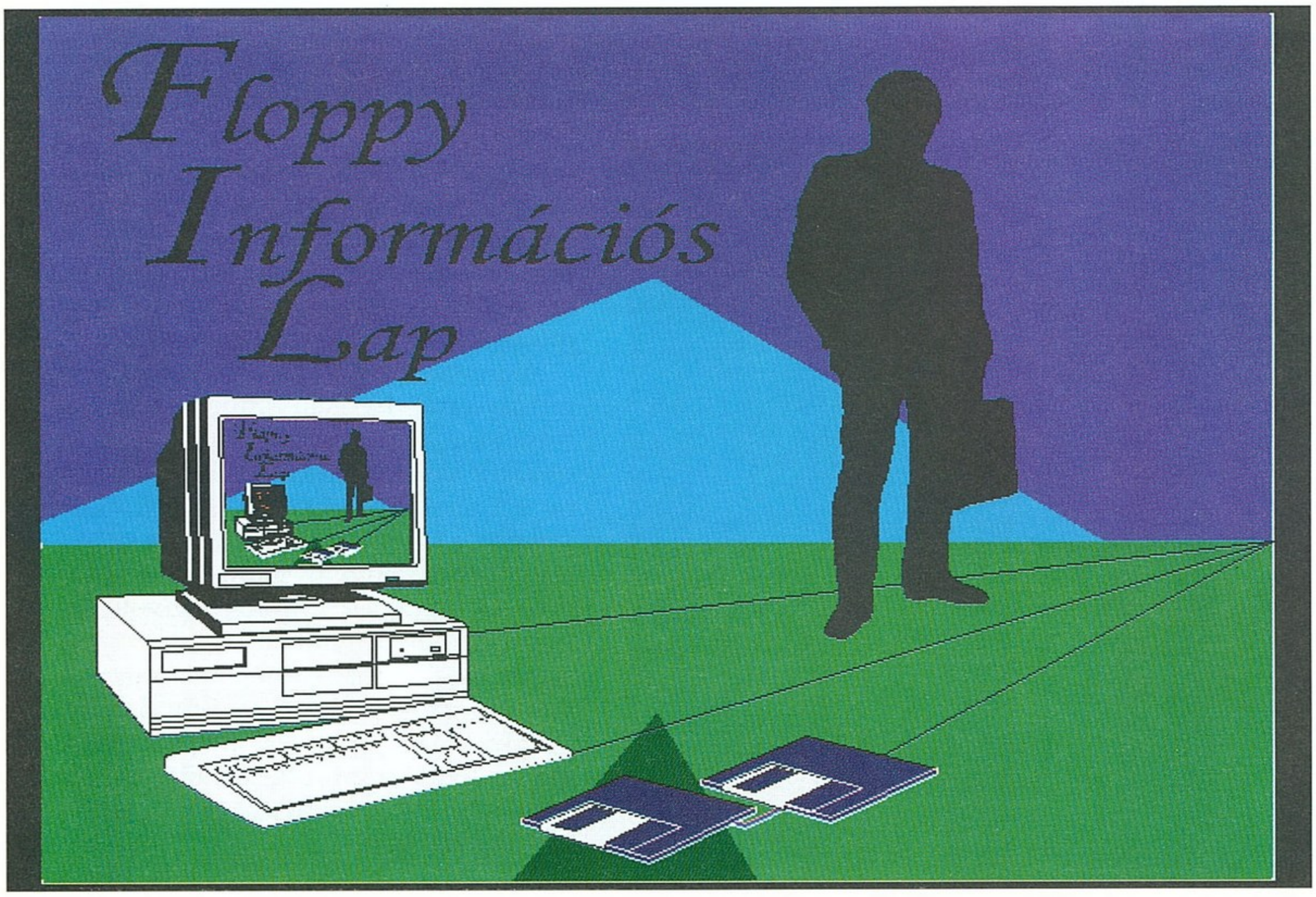
— Az MNB információi.

— Nemzetközi vásárnaptár országonkénti bontásban.

— IKM információk.

A lap *számítástechnikai* kivitelére az jellemző, hogy nem törekszik túlcicomázott kivitelre, hanem a tömény (és jól összepréselt) tartalmi információ az erőssége.

Varga János



Silver Wolf Desktop

Alternatív ablakok

A Windows 95 megjelenésével a személyi számítógépek majdnem kétharmada azonnal korszerűtlenné vált... ...volna, ha más cégek nem tudnának kiutat mutatni a Bill Gates („és társai") által állított csapdából.

A domináns tábor célja, hogy az asztalra az alkalmazók egyre nagyobb teljesítményű számítógép-szörnyetegeket vásároljanak, tekintet nélkül arra, hogy anyagilag bírják-e ezeket a terheket. A másik irányzat szerint (az Internet nagy öregjének, Nicklaus Negropontenak a szavait idézve): a Bill Gates—Intel összeesküvés ellen fel kell lázadni. Olcsó eszközöket kell az alkalmazók kezébe adni, amelyekkel könnyedén érhetik el a hálózatokat.

Ez utóbbi felfogással hardveroldalon a legnagyobb meglepetésre a jelenlegi adatbázis-nagyhatalom Oracle jelent meg: 500 dollár alatti gépet akar gyártani. Szoftveroldalról pedig a kelet-európai szoftverügyekben járatos szakértő, Ester Dyson támad, szintén vitára ingerlő jelszóval: legyen minden szoftver ingyenes!

Mac-PC szimulátor

Kell-e ekkora „bombasztikus" bevezetőt kanyarítani egy kis shareware szoftver, a Silver Wolf Desktop (SWD) ismertetése elé?

Igen. Ez a program ugyanis éppen azt a szimpatikus irányzatot példázza jól, amely megpróbálja megőrizni az alkalmazók beruházott eszközeinek értékét és használhatóságát, a programok tudásának növelésével párhuzamosan.

Milyen irányba mutat? Bill Gates ablakaitól a Macintosh ablakai felé, amelyeket a Windows 95 az Apple szerint már majdnem elért — 10 éves Mac-technológiával „ajándékozva meg" az Intel-PC világot.

Programkezelő a sarokban

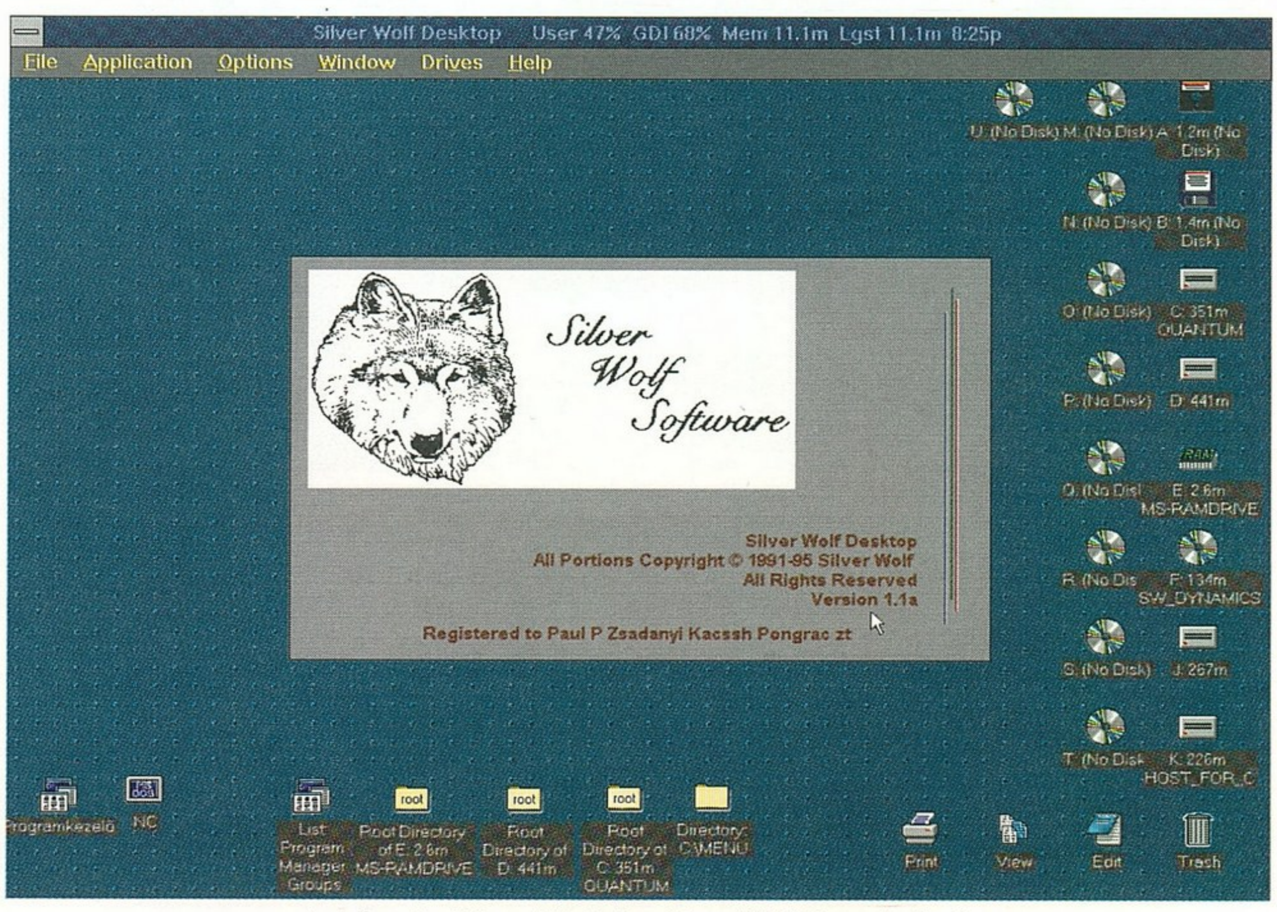
Mint a mellékelt 1. kép mutatja, a Windows 3.1x Program Manager (Programkezelő) programját a Silver Wolf Desktop mellett nyugodtan a sarokba térdepeltethetjük kukoricára, amiért oly sok éven át kínzott bennünket idétlenségeivel. Nézze végig, mit csinál helyette a Program Managert és File Managert együttesen helyettesítő, és Mac-képernyőt varázsló SWD, amelyet installálás után betettünk a Rendszer és a Start csoportba! Az utóbbi folytán automatikusan indul. Ha néha véletlenül ledöglik — többnyire a WinDog underground miatt —, akkor a Program Managert arra a szégyenteljes feladatraösztökélhetjük (ha még egyáltalán feléled), hogy hívja be újra az SWD-t, amely ismét a sarokba küldi, hogy ne foglalja el a drága képernyőfelületet.

CIS Shareware Registration

Azért legyünk igazságosak, néhány konfigurációs funkció még mindig a Program Managerrel végezhető el a legjobban. Talán szólni kéne a Silver Software csoportnak, hogy varázsolják bele az SWD-be a még hiányzó funkciókat. A szerzőnek erre kevés energiája maradt, pedig csak egy e-mail levél volna a CompuServe-ön át — tekintve, hogy a szoftvert onnan töltötte le az első változatában, és ott regisztráltatta (GO SWREG).

Az utóbbi lehetőség nagyon kényelmes. Nálunk eddig külön cirkusz volt, ha valaki regisztráltatni akart egy szoftvert. Ki kellett töltenie valami papírt, majd azt el kellett küldenie a szerzőnek. Eddig még rendben, de elküldheti-e a regisztrációs díjat valamilyen ninivei rúpiában? A CompuServe SWREG fórumának a segítségével ez a dolog annyira leegyszerűsödik, hogy ha kitöltünk egy online formalapot, azt a rendszer elküldi a szerzőnek, a számlánkra pedig egyszerűen ráterheli a regisztrációs díjat, s mi forintban fizetjük, egyéb szolgáltatás gyanánt.

Ez a módszer talán törvényeket sért, mind nálunk, mind Amerikában, de nagyon jó, hogy a hatóságok szemet hunynak fölötte. Igen kellemes az az érzés, hogy hipp-hopp, online viszonyba kerülhetünk a program szerzőjével, aki folyamatosan küldi a változások leírását e-mailen át, és ugyanazon a csatornán konzultálhatunk vele. Esetleg a regisztrációs kódot is így kapjuk meg.



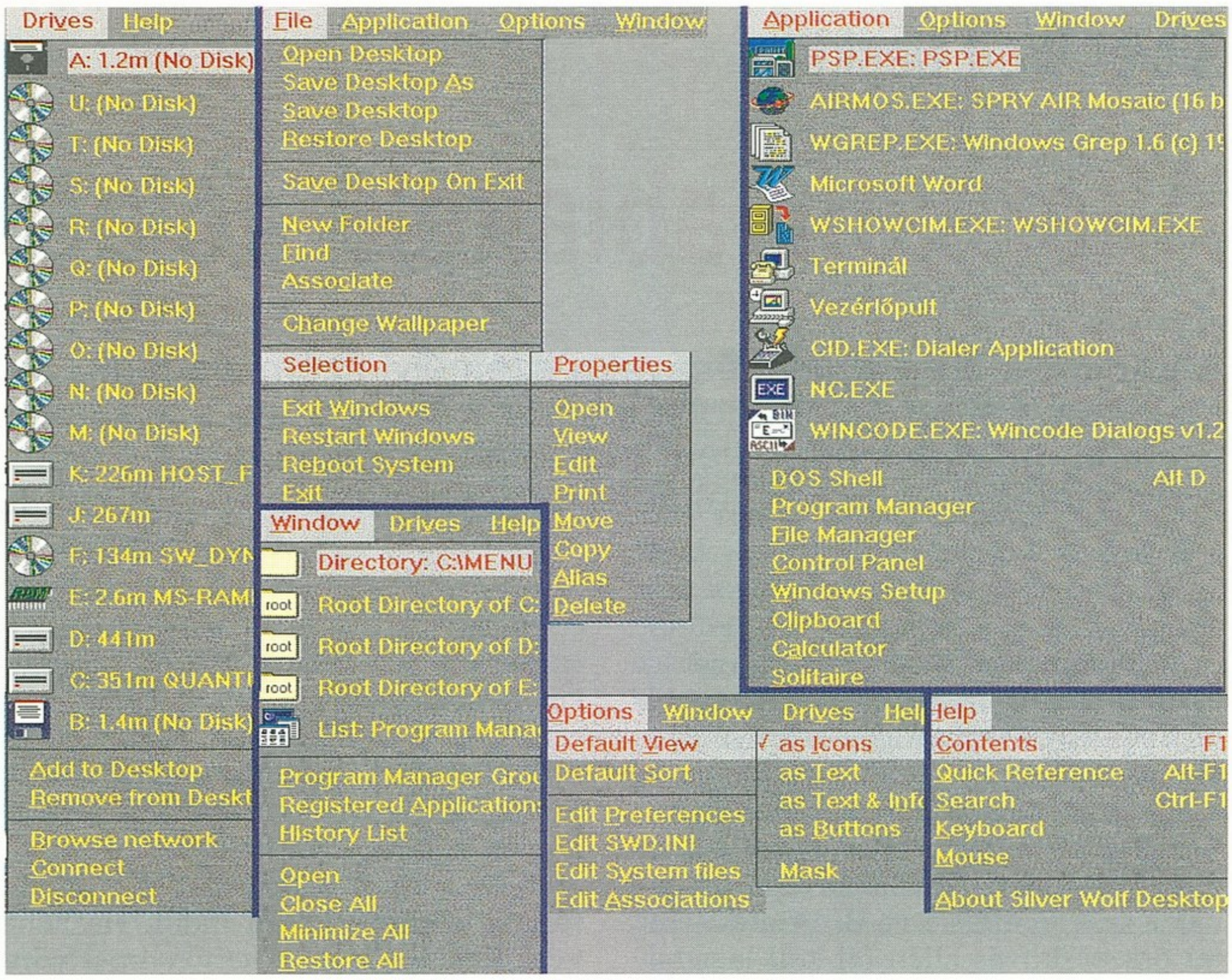
1. kép. Az Silver Wolf Desktop (SWD) asztallapja

Sőt. Attached File-ként (levélhez kapcsolt állományként) az új változatot binárisan is átküldheti. Ezt a CIS-ben még uudecode-olni sem kell, mint az Interneten át érkező bináris állományokat, mert a CIS alatt nem egy buta 7 bites TCP/IP protokollos hálózat ül, hanem OSI-kompatibilis DECnet. Amúgy az SWD-t postán át, floppyn is megkapja az ember egy kis doksi kíséretében.

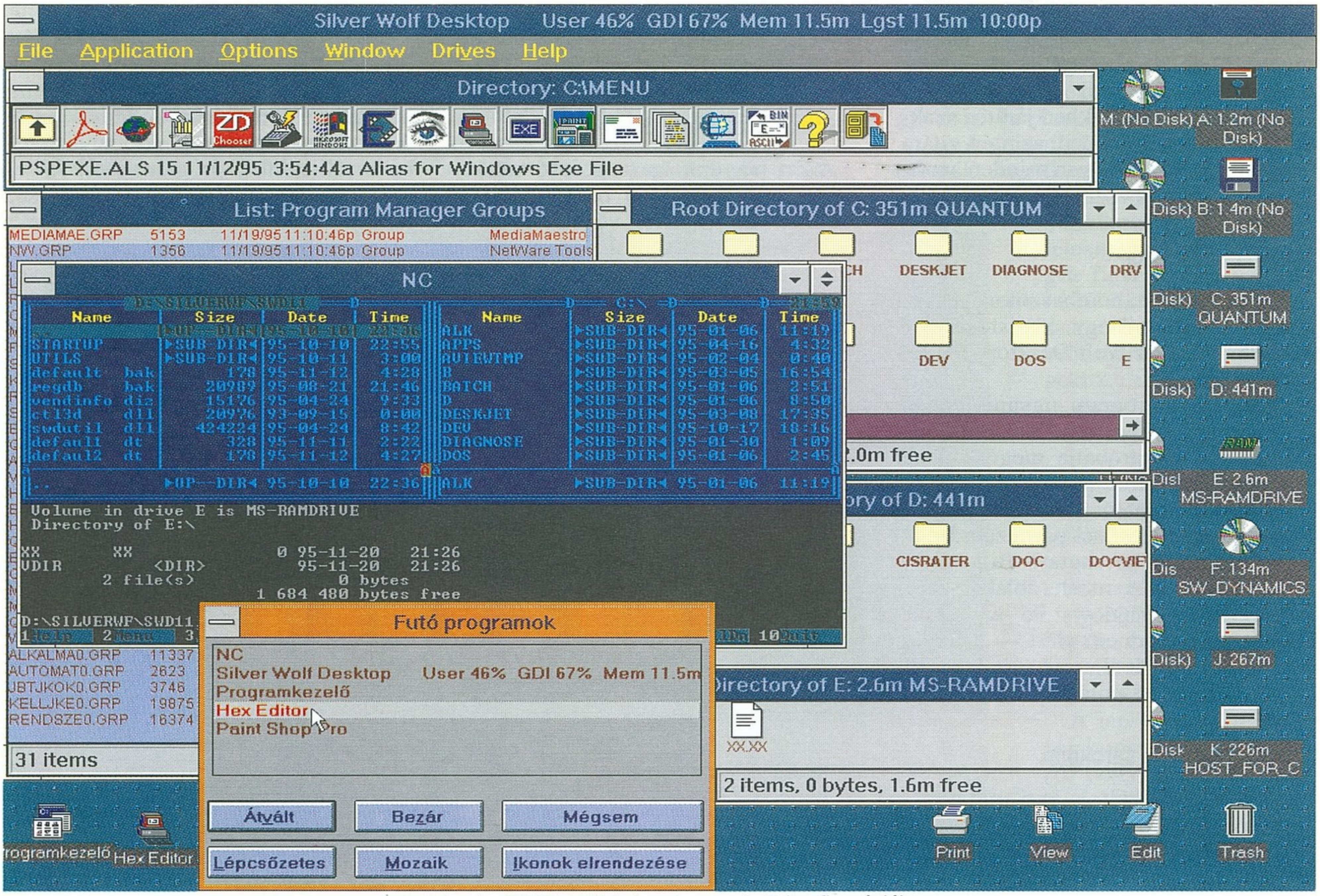
A regisztrációs díj 39 dollár volt, ehhez jött még 8 dollár szállítási költség. (Az SWREG díjak ritkán érik el a 100 dollárt. Ez egyébként azonos a CIS-en a heti korláttal, ami után automatikusan leállítják a felhasználó ID-jét, nehogy egy betörő megrabolhassa a tulajt, viszont a tulajnak is gondjai támadnak, ha sokat rendel.)

Egy kép többet mond...

A mondanivaló java részét képekbe sűrítettük. Az 1. képen egy mozdulattal ikonba küldtük az összes útban lévő ablakot, hogy bemutassuk az SWD alapképernyőjét. Kis csalás, hogy háttérként a WfW Rivets raszterét választottuk ki, de már az SWD File menüjéből mindig egyszerűen aktivizálható háttérválasztó funkcióval. A színeket még a Windowsban konfiguráltuk, hogy ne égessék ki a szemünket a



2. kép. Az SWD menümontázsa



3. kép. Az SWD saját igényekre szabott konfigurációja

FullWhite ablakok, meg hogy egy kis változatosságot mutasson a képernyő (alapzöldünk megdöbbentő módon a W95-ben is előkerült, így választásunk nem lehet minden logika nélküli). Az SWC, a konfigurátor további paraméterezéseket tesz lehetővé, de néhány szín már csak a Rendszercsoport színkiválasztójával határozható meg.

A 2. kép bemutatja az SWD összes fő-, és majdnem minden almenüjét, montázsként úgy, hogy ne sok maradjon le az eredeti részekből. A File menü érdekessége, hogy külön funkció az SWD, és külön a Windows leállítása. Aki akar, bármikor visszarohanhat az Ablak ablakaihoz az SWD lelövésével. A tapasztalat szerint nem fog kevésbé gyakran lerobbanni a rendszer. Ami a többi fejlécmenüt illeti, az Application tartalmazza a legutoljára behívott programok ikonjait, amelyeket gyorsan újraindíthatunk. Ezen a menün mindig ott vannak a leggyakoribb rendszerelemek (a Programkezelő is).

A Drives menü szintén unikum, mert kiválaszthatjuk, melyik X: egység legyen rajta a listán, melyik ne. Rájuk kattintva megnyílik egy Folder/Directory, szóval egy könyvtárablak (unixosan inkább katalógus, Mac-esen dosszié). Ennek tulajdonságait elemezhetjük, megváltoztathatjuk stb., ha a jobb egérgombbal rákattintunk a fejlécére. Ugyancsak ilyen repülőmenü ugrik elő a jobb gomb megnyomására, ha egy állományra kattintunk rá. Ha az asztallapra kattintunk kétszer a bal gombbal, akkor előugrik a Windows programváltó ablaka, mint az a 3. kép alján is látható.

A 3. képen visszaküldtük az ablakokat a helyükre a Windows menü egyik parancsával. Érdemes elböngészni a képen. Némi csalással nem a szokásos 640*480-as, hanem 1024*768-as felbontású a megjelenítés. Különben aligha férne el ennyi információ rajta. (Sajnos, az alkalmazók hanyagságból nem mindig állítják be a lehetséges magasabb felbontást a gépükön, tehát feleslegesen rongálják a szemüket, és kihasználatlanul hagynak olyan lehetőségeket, amelyeket pedig a drága monitor és a képvezérlőkártya révén már megszereztek. A márka nélküli gépek tulajdonosai néha nem is tudják, mire képes a gépük, pedig csak ki kellene próbálniuk...) Amire szívesen felhívjuk a figyelmet a 3. képen: a menüsor alatt van egy saját készítésű menünk is. Egy képablak tárgyait ugyanis kérhetjük ikonos, szöveglista, szöveg és információs lista és végül nyomógomb alakban. A C:\MENU katalógusunkat egy gombsávnyira konfiguráltuk. Azután a legkülönbözőbb helyekről másoltunk bele programokra hivatkozó Alias (álnév) tárgyakat. A tárgyak hosszú neve az SWD-ben egyébként 128 karakteres lehet, ami kivezet végre a 8+3-as névnyomorból. Ez persze a Mac-en már rég természetes volt, meg a mai Unixokban is.

A farkas megeszi...

Ha azt mondtuk, hogy a mai asztali PC-s Unixokkal jobban lehet DOS-ozni, mint az igazival, akkor azt is elmondhatjuk, hogy itt-ott az SWD-vel Mac-OS-ezni lehet jobban, mint az igazival. Ennek kulcsfigurája a nem egy-, hanem maximálisan háromgombos egér. De a billentyűzetparancsok se kutyák. A Mac-világból érkezőknek a leírás ad némi segítséget, hogy miként igazodhatnak el a számukra egyébként világosnak tűnő SWD-asztalon a Mac-en megszokott technikával. A drag-and-drop (húzd oda és rakd rá) mindenesetre kiválóan működik. A ZIP állományok read only (csak olvasható) ablakokként is megnyithatók. Az SWD az SWC konfigurátorral együtt sem foglal 2 Mbájtnál több helyet a diszken. 4 Mbájt RAM-ban már fut, de azért minimum 386-os processzor kell neki. A 286-osokon tehát az SWD sem segít. Sajnos a 286-os az Intel első, kicsit még éretlen multitaszkos processzora volt, amelyen tényleg nem lehet segíteni (hogy az XT-k monotaszkos 88/86-osairól ne is beszéljünk).

Még egy utolsó — meglehet, hogy elfogult — megjegyzés: a szerző az SWD-t hónapok óta stabilan használja, és nem kívánja vissza az MS Windows eredeti ablakait.

(Silver Wolf Software: Mark Daly CIS: 71441,1637; Web: http://www.silverwolf.com/swi/sws/.)

Zsadányi Pál

A Unix is nyert vele

Az SWD ismertetése azért is érdekes, mert egyrészt a Mac OS is jobb, mint a Windows, másrészt az SWD a W95-alternatívák között akár jelentősebb helyet is kiharcolhat magának. Később a W95-alternatívákat azzal folytatjuk, hogy az asztalon a Windows 95-nek már túl gyenge hardverekre jól fejlett asztali interfészekkel felszerelt Unixok is rendelkezésére állnak, olyan DOS/Windows emulátorokkal, amelyekkel jobban lehet „doszozni", mint az igazival. Az árakat tekintve az elérhetők közé tartoznak a Linux-alapú rendszerek, de profi igények is kielégíthetők a Novell UnixWare és az SCO OpenServer asztali változataival, persze nem olcsón. (Az utóbbi kettő éppen most nő egymásba, amint ezt novemberi számunk UnixWare-es cikke megvilágította.)

A Unixokkal a Windows 95 miatt vásárolt nagyobb teljesítményű gépekből is jóval több hozható ki, mint az új Ablak-világrendszerrel, ráadásul stabilabb adatvédelmi és rendszertechnikai paraméterek mellett.

...Toldd meg egy lépéssel!

QEMM 8.0

Típus: Univerzális memóriamenedzser.
Gyártó: Quarterdeck Corporation.
Minimális hardverkövetelmények:
Windowst vagy Win95-öt futtató PC-n
4 MB RAM, csak DOS-os PC-n
1,25 MB RAM,
4,5 MB háttértár-kapacitás.
Támogatott operációs rendszerek:
PC vagy MS DOS 3.3 és efölötti
verziók, DR DOS 6 és Novell DOS 7.
Ajánlott listaár: 16 000 + áfa,
upgrade: 9000 + áfa.

A Microsoft MS DOS 6.22-es, utolsónak kiadott dobozos operációs rendszere sem büszkélkedhetett azzal, hogy virtuóz módon bánik a RAM-mal. Tehát létjogosultsága volt a különböző memóriamenedzser megoldásoknak. Ezek közül három méltán vívott ki magának nemzetközileg is elismert sikert, az eladott példányszám sorrendjében a következők: QEMM, utána sokáig szünet, majd a Helix Software NetRoomja és Multimedia Cloakingja (mellesleg ennek lebutított változatát hívják az MS DOS 6.x-ben MemMakernek), végül a 386MAX (itt már a név felett is kissé eljárt az idő...).

Önálló diszkkezelő alrendszerrel ellátott operációs rendszerek esetén (lásd MS Windows NT, IBM OS/2 Warp, Unix, Linux stb.) természetesen szó se lehet ilyen memóriakezelésbeli toldozásról-foldozásról. Viszont a Microsoft oly népszerű Windows 95-je egy felemás állapot. Ott lapul még laikusok számára is felismerhetően egy DOS, nevezetesen az MS DOS 7.0. Így maradt még egy kis hely a pályán a QEMM és társai számára, igaz, már csak az oldalvonal mellett egy keskeny sávon, de még a pályán belül, jelezvén, hogy a Microsoft itt sem alkotott 100 százalékig tökéleteset.

A Win95 memóriakezelése olyan mértékben kifinomult és automatikus, hogy szinte kár belepiszkálni. A DOS promptnál nyer az ember maximum 4 KB-ot, no de enélkül is vígan elérhető a 626 000 szabad bájt. Ehhez kapunk még vagy négyezret. Grafikus felületen gyakorlatilag ikonról (bocsánat, Shortcutról) hívható a QEMM Setup, a Manifest nevű szemléletes tesztprogram és a Win95, QEMM és MagnaRAM hármas hivalkodó, bár effektív haszonnal nemigen kecsegtető statisztikai képernyője.

Igazából a Windows 95 alatt a következő okok miatt lehet szükség a QEMM-re:

— Egy nagyobb program (pl. CorelDRAW, MS Access 95, Excel 95 stb.) elindításához kevés a fizikai RAM-unk. A MagnaRAM addig trükközik, mígnem majdnem minden a duplájának látszik, és a memóriafaló programok ily módon általvervén mohón elindulnak.

— Megbújik még valahol a merevlemezünkön a régi, Win95 előtti DOS és a kimustrált öreg Windows, s azt szeretnénk, ha itt minden flottul menne. (Speciális installálási lehetőség, az egyik technikai tanácsadó fájlban lelünk csak rá, hogy hogyan is tegyük, de megéri.)

— Megszokásból, illetve egyszerűen csak nosztalgiából.

Régi Windows használata esetén hozzájön még ezekhez a megnövekedett megbízhatóság és a MagnaRAM által felpuffasztott Resource és GDI terület is.

A csomag, mint azt már többször is említettük, magában foglalja a MagnaRAM-ot is, amely lényegében azt teszi, mint a Stacker a harddiszk-területtel. Többszörözi. Az üres, adat nélküli RAM-ot az egyszerűség kedvéért megduplázza, ha adat kerül belé, a tömörítési arány általában romlik, de speciális esetben (redundáns, ismétlődő mintákat tartalmazó adathalmaz) az arány 2-nél is nagyobb lehet. A lényege: a lufi módjára felfújt RAM kívülről nagyobbnak látszik, később telik meg, később fordul az operációs rendszer a harddiszken allokált virtuális memóriához, így mivel a RAM-ban minden kb. két nagyságrenddel gyorsabban zajlik le, a MagnaRAM időt takarít meg számunkra, ezt egy statisztikában szerényen, másodpercekben mérve ki is írja a státusképernyőre. Mindamellett „végteleníti" a resource terülelet is, elébe menvén az „Out of memory" üzeneteknek. Jobbnak mondható, mint a RAM Doubler vagy a SoftRAM.

Konklúzió: a termék 2 hetes tesztünk alapján korrektnek, mi több, hibátlannak bizonyult. Nem ehhez szoktunk, a kerek verziószámú kiadásokba eddig rendszeresen több apró bug rekedt, ezek rendszeresen kirajzottak (azaz elszálltak a programmal egyetemben). Ilyenkor legalább 2-4 hónapot kellett várni az igazira, amely verziószámban néhány addicionális tizedet vagy századot jelentett. Most nincs mire várni, ezt a változatot készen hozták forgalomba. Aki teheti, vegye meg.

Delrina WinFax Pro 7.0

Típus: univerzális faxszoftver.
Gyártó: Delrina
(már a Symantec többségi tulajdona).
Minimális hardverkövetelmények: 8 MB RAM,
16 MB háttértár-kapacitás, faxmodem.
Támogatott operációs rendszer: MS
Windows 95
Ajánlott listaár: 19 600 + áfa

A kanadai faxszoftverkirály céget, a Delrinát is elérte a végzete, a falánk Symantec, amely viszont a segédprogramok monopóliumára törekszik. Egyelőre még nem működik zökkenőmentesen a nász, mert a Delrina megtartotta független értékesítési csatornáit és „dizájnját".

Kezdhetnénk azzal, hogy e csomag tartralmával azokat a réseket tömi be a Wfax, amelyeket a Microsoft felejtett benne (a nagy sietségben) a Windows 95-ben, hiszen annak része az MS Fax. Lássuk tehát:

— Még barátságosabb faxwizard.

— A Xerox cég Textbridge nevű OCR (Optical Character Recognition — karakterfelismerő) szoftvere az érkezett szöveget is tartalmazó állományok editálhatóvá alakítására.

— Munkánk megszakítása nélkül a háttérben küld és vesz fax- és e-mail(!) üzeneteket, alapos (mintegy egyharmadára zsugorító) automatikus szoftvertömörítési technikával, ami jótékony hatással lesz telefonszámlánkra, különösen ha távhívunk.

— A kényelmes Delrina CommBar a képernyő alján mindig tájékoztat faxmodemünk állapotáról, az adás/vétel

előrehaladásáról, grafikusan ábrázolva. Egy kis piros bigyó kezd el villogni a jobb szélen, ha új üzenet jött.

— Beépített hívásszétválogatás (Call Discrimination) tesz különbséget a bejövő bináris adat és a faxüzenet között.

— MAPI (azaz Microsoft Exchange) kompatibilitás, amelynek révén az érkezett fax és e-mail üzenetek egy központi helyről kezelhetők, nézegethetők, kinyomtathatók, rendezgethetők.

— Autoforwarding: ha nem vagyunk otthon, de megvan az elérési telefonszámunk, a WFax utánunk küldi az otthonra érkezett üzenetet.

— Amennyiben voice opcióval is ellátott modemmel rendelkezünk, további szenzációs szolgálatásokat élvezhetünk: a TalkWorks (ezt később jelenteti meg a Delrina) segítségével hangpostaládákat létesíthetünk, illetve egy intelligens üzenetrögzítővel gazdagodhatunk. Fax-on-Demand szolgáltatást is biztosíthatunk kifelé: rossz idők járnak a faxbank-üzemeltetőkre: ezentúl minden WinFax Pro 7 tulajdonos (voice mail chipes faxmodemmel!) saját faxbankot robbanthat...

— Caller identification — de ez a szolgáltatás kis hazánkban postailag nem engedélyezett, ezért nem is fordítom le magyarra.

Multimedia IQ Test CD

Típus: Önbizalomnövelő(!?) szoftver.
Gyártó: Virtual Entertainment.
Minimális hardverkövetelmények:
4 MB RAM, 386-os processzor, SVGA, hangkártya avagy Macintosh gép 256 színnel, 4 MB RAM-mal.
Szoftverkövetelmény:
Windows 3.1 (Win95 is jó),
MAC: System 6.0.7 vagy System 7.x.
Ajánlott listaár: 4800 + áfa,
sőt More IQ Tests CD: 5400 + áfa.

Néhány sallang, reklám és gyermekpszichológusokkal készített riport mellett a CD-n megtalálható lényeg:

150 kérdéses intelligenciateszt. Megoldási idő: 40 perc. Többszörös választás, azaz a helyes felelet megadása a négy megadott lehetséges alternatíva közül. A három fő kérdéscsoport:

1. Ritkábban használt angol szavak, idegen szavak jelentésének kitalálása.

2. Látszólag pofonegyszerű, két-három mondatos szöveges számtanfeladványok a 100 és 1000 közötti tartományban.

3. Több egymásra rakott kocka vagy hasáb számának meghatározása (a térszemlélet és az absztrakció elllenőrzése).

A végén kiértékelés, tudományos magyarázatok. Angolul tudni kell. E sorok szerzője 40 perc alatt 133 kérdésig jutott el, az IQ-ja 128 lett. Főleg az 1-es típusú kérdésekkel időzött el. Magyarul talán okosabbnak bizonyult volna...

Quick View Plus 3.0

Típus: Univerzális állománymegjelenítő.
Gyártó: Inso Corporation.
Minimális hardverkövetelmények:
4 MB háttértár kapacitás.
Szoftverkövetelmény: Microsoft Windows 95.
Ajánlott listaár: 12 000 + áfa.

A CD-s Windows 95-nek része a Quick View szolgáltatás, az Explorerbe beépített állománymegjelenítő. (Megjegyezzük, ha az embernek még Norton Navigator 95-je is van, a Quick View az Explorer ablaka alá vagy mellé „tapad", mint állandó fájlkukkoló ablak.)

A Quick View tulajdonképpen a Quick View Plus kistestvére. A terjedelmes listát a QVP meg tudja jeleníteni a képernyőn, és ki is nyomtatja, másolja, clipboardra teszi részben vagy egészben stb. A lista igen tekintélyes következő: ASCII text, DEC WPS, Word for DOS, Word for Win. 1.0-7.0, RTF, Ami Pro, HTML Internet, MacWrite, MS Works, Q&A, WordPerfect, WordStar, XyWrite, Lotus 1-2-3, Symphony, Excel, Quattro Pro, Access, dBase, Fox, Paradox, AutoCAD DXF, BMP, CGM, CorelDraw, DCX, EPS, GIF, TIFF, Harvard Graphics, HPGL, Windows Icon & Cursor, MacPaint, PCX, Powerpoint, RLE, Targa, WMF, WPG, Windows DLL és EXE, DOS EXE, PKZIP, TAZ, UUE Internet, TAR és még sok más egzotikus formátum is.

OLE-objektumokat, fontokat, táblázatokat is megjelenít teljesen natív formában. Jobb egérklattyra mindig készen áll. Automatikusan felismeri az elterjedt fájlformátumokat. A Win95 Find-jával kooperálva a találatokat eredeti környezetükben tekinthetjük meg. Ugyanígy jótékonyan betüremkedik a Microsoft Exchange-be is (ez főleg Mac-es géppel történő adatcsere esetére, valamint e-mailhez csatolt állományok megjelenítésére használható nagy sikerrel).

Herczeg József

Kiegészítés az Új Alaplap 95/6. számában megjelent listához

Magyar CD-ROM-ok diszkográfiája

Teljes szövegű adatbázisok

Bertelsmann LexikoDisc 2.0
Bertelsmann Lexikon Verlag;
Hypermédia Systems, 1993.
ISBN 3 570 11 020 6

Népszabadság 1994. II. félév
Népszabadság Rt.;
Hypermédia Systems,1995
Ára: 15 000,- Ft (K)

Nyelvoktató programok, szótárak

Anyanyelvi könyvespolc
A magyar helyesírás szabályai,
Helyesírási kéziszótár,
Idegen szavak szótára,
14 ezer szavas
értelmező szótár
Akadémiai K.; Scriptum, 1995.
Ára: 10 000,-Ft

Német-magyar hangosszótár
Akadémiai K.; Scriptum, 1995.
Ára: 15 000,-Ft

ClipDIC English with Coursebook
Kulcs az élő nyelv megértéséhez
Profi-Média, 1995.
Ára: 5600,-Ft (K)

Hello
Angol kiejtésiskola
Profi-Média, 1995.
Ára: 5800,-Ft (K)

Nyelvmester
Nyelvoktató CD: Középhaladó angol
Vision-X, 1995.
Ára: 8750,-Ft (V)

Nyelvmester
Nyelvoktató CD: Középhaladó német
Vision-X, 1995.
Ára: 8750,-Ft (V)

Multimédia adatbázisok

abc CD
Szinva Net;
CD Multimédia Szoftverház, 1994

Hungarian phone card catalog '95
Color Plus; MATÁV, 1995.
Ára: 7500,-Ft

Parcel 301: Jovánovics
ABCD; Ikon, 1995

Manóka-land
Játékos képességfejlesztő
oktatóprogram kisiskolásoknak
Profi-Média; Ábécéskönyv 1995
Ára: 6150,- Ft

Az ellopott kódex titka
CD-ROM mesejáték
Minor, 1995
Ára: 4590,- Ft (K)

Indul a bakterház
Com-Ser; Televideo, 1995
MPEG Video

Hat nyelvű Európa szótár
CyberStone, 1995.
Ára: 6900,- Ft (A)

Ebciklopédia
Automex, 1995.
Ára: 5000,- Ft (A)

Interactive „PUSzTA" Sex
CyberStone, 1995.

KRESZ
Oktató- és vizsgáztató program
Pictron; CD4, 1995.
Ára: 6500,- Ft (K)

Alfred Brehm: Az állatok világa
Kossuth; Vision-X, 1995.
ISBN 963 09 3810 3
Ára: 8900,- Ft

Bartók Béla
Magyar Rádió;
Hypermédia Systems, 1995.

Egyéb CD-ROM-ok

Nemzetközi szabadalmi osztályozás
5. kiadás, 1990.
Arcanum; Tudorg; Gloria, 1990.
A kereskedelemben már nem
kapható

Őskori iparvidék a Bakonyban
Magyar Nemzeti Múzeum;
Com-Ser, 1995.

Korongvilág
CD Archive, 1995—
Megjelenés félévenként
Ára 990,- Ft/szám

Music Vision Part 1
Vision-X, 1995.
Ára: 2500,- Ft

Keszoft shareware
1000 DOS és Windows shareware
program. Keszoft, 1993—
Kb. évente jelenik meg,
a 2. kiadás 1994-ben,
a 3. 1995-ben jelent meg

A listára azokat a CD-ROM-okat vettem föl, amelyek 1995. május 15 — november 30. között új címként jelentek meg (a periodikusan megjelenő CD-k újabb kiadásait nem számítva), illetve jutottak tudomásomra.

Tószegi Zsuzsanna

Rajz, majd valóság

Legalább kipróbálható...

Az elektromos, elektronikus tervezés legalább annyira nem nélkülözheti a CAD adta lehetőségeket, mint a gépészet vagy az építészet. A CAD elektromos területeken való alkalmazásaiként készülnek az EDA szoftverek. (EDA = Electronic Design Automation = elektronikai tervezés-automatizálás.)

Ellentétben más tervekkel, az elektronikus áramkörök tervei már magával a PC-vel meg is valósíthatók. A megtervezett áramkör logikai része ugyanis a tervezőprogramból való áttöltéssel realizálható egy programozható logikai eszközben, amit általában csak a PLD (Programmable Logical Device) rövidítéssel illetnek. A PLD — mint a gyurma — olyan alakítható „test", amely a logikai áramköri lehetőségek végtelenjét hordozza magában.

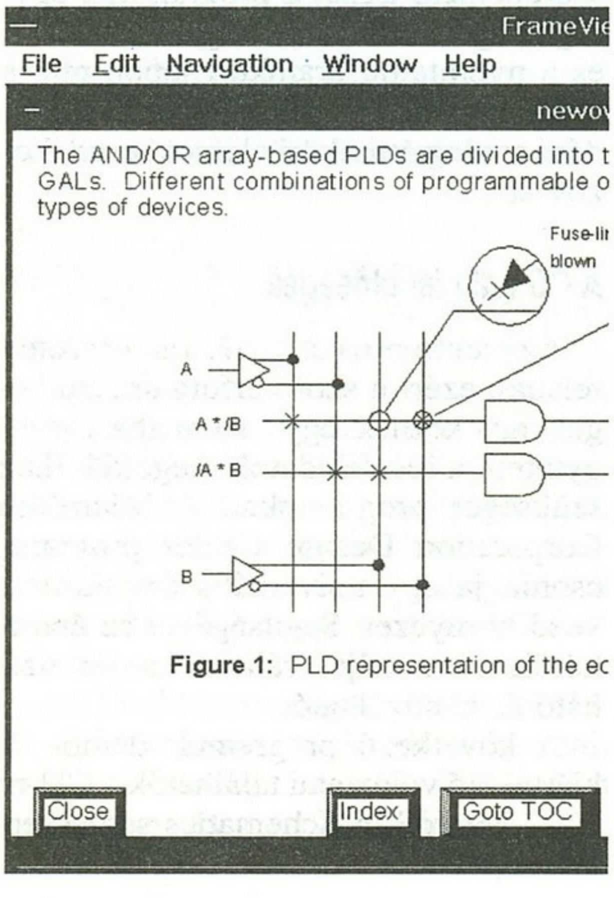
Eltérések az elektromos és a geometrikus CAD között:

— Itt elegendő 2D-ben ábrázolni (X-Y).

— A csomópontok és vonalak csak a rácspontokon (grideken) lehetnek.

— A rajzelemek elektromos tulajdonsággal bíró szimbólumok.

— A rajzi szimbólumok és az elektromos paraméterek összerendelése a már szinte szabvánnyá lett SPICE szerint ún. helyettesítő modellek alapján történik (SPICE = Simulation Program with Integrated Circuit Emphasis).



Az integrált áramkörök megjelenése előtt minden felhasználó saját maga tervezte meg sok-sok tranzisztorból és ellenállásból felépítve a számára fontos logikai kapcsolást. Azután az IC-gyártók elkezdték ontani a sokszor ismétlődő elemi feladatok megoldására alkalmas kapuáramköröket. A Texas TTL logikai családja, az SN 74XXX sorozat világszabvánnyá vált. Ezekből most már könnyebb volt megtervezni és elkészíteni a logikai kapcsolásokat. Készültek is ez időben sokkártyás, nagyméretű vezérlődobozok, műszerek stb. Természetesen kisebbek, tömörebbek voltak tranzisztoros elődeiknél.

Ahogy az IC-gyártás technológiája finomodott, lassanként megjelentek a kínálatban az egy chipen sok logikai funkció megvalósítására alkalmas ún. elrendezések (arrayk). Ezek a PLD-k.

„Csináld magad!"

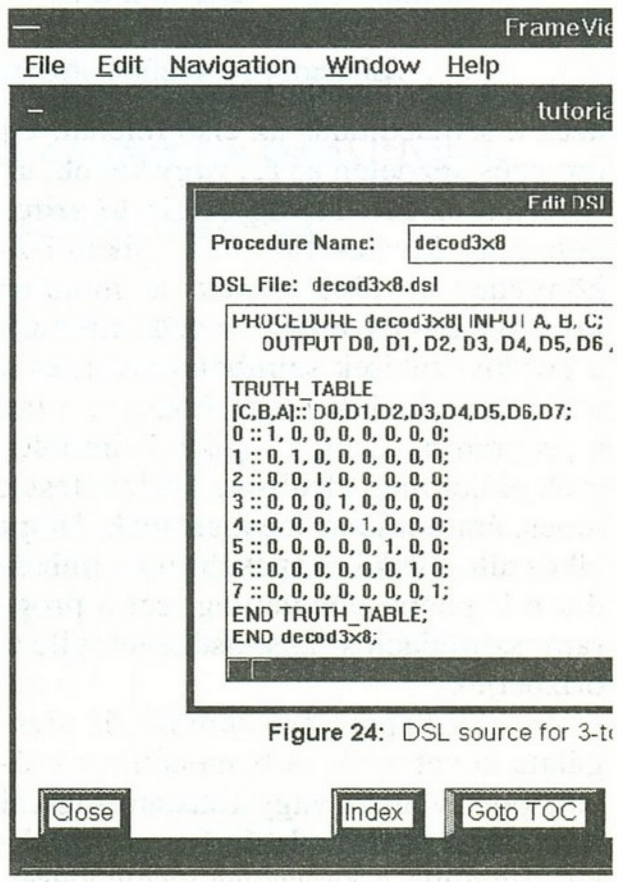
Saját igényeink szerint készíthetünk logikai áramkört a PLD szintetizáló (PLsyn) EDA programmal. A MicroSim Corporation Design Center teljes tervezői programcsomagja, illetőleg ennek Evaluation változata CD-n érhető el Magyarországon is. Ez a programcsomag tartalmazza többek között a PLD szintetizáló lehetőséget is.

A teljes változat természetesen nagy, összetett feladatok professzionális megoldására való, de az Evaluation változat segítségével kisebb vállalkozások vagy oktatási intézmények is megízlelhetik a berendezésorientált áramkörkészítésben rejlő lehetőségeket. Az alkalmazási szférák közül elegendő az egyedi védelmi rendszerek kidolgozásának lehetőségét megemlíteni.

A PLD az egyedi igények kiszolgálására alkalmas, merthogy a felhasználónak kell és lehet az IC-beli kapcsolást véglegesítenie, azaz „beleégetnie". A beégetés kifejezés azért jogos, mert bizonyos változatokban a végleges belső összeköttetéseket ún. biztosítékok (fuse-ok) kiolvasztásával kell kialakítani. A PLD IC maga kapukat, flip-flopokat és kimeneti áramerősítőket tartalmaz. Így ma már egy nagyobbacska IC-tokban megvalósítható olyan összetett logikai feladat, amely korábban több tenyérnyi nagyságú nyomtatott áramköri kártyát igényelt.

Miért PLD?

A PLD eszközök segítségével könnyen, gyorsan, takarékosan oldhatjuk meg fejlesztési problémáinkat. Nem kell ugyanis raktáron tartanunk vagy megrendelnünk a sokféle kis bonyolultságú IC-t, elegendő néhányféle PLD-t beszerezni, az igény jelentkezésekor ezekből egyet gyorsan beprogramozni, és máris kész az áramkör tesztelhető, bemutatható prototípusa. Amennyiben ez bevált, ugyanebben a változatban természetesen sorozatban is készülhet tovább. Egy hozzávetőleges összehasonlítás a kis bonyolultságú katalógusáramkörök és a PLD között:



Méretviszonyok: 1 db MACH 231 mérete: 1,19" x 1,19" = 30 mm x 30 mm = 900 mm^2, ami elválasztó térközt nem számítva 8-9 darab 14 kivezetéses, kis bonyolultságú, DIL tokozású IC helyfoglalásával egyenlő. (1 db 7400-as alapterülete = 6x20 = 120 mm^2.)

Fogyasztás: A MACH 231 150 mA-es áramfelvétele kb. 9 darab 74LS00-

éval egyenlő (a 74LS00 áramfelvétele 4x4,4 mA = 17,6 mA 5V-ról).

Sebesség: A MACH sorozat ún. High/speed CMOS, késleltetési ideje 10 ns (a TL LS sorozata is 10 ns-os).

Ebből a kis számpéldából is látható, hogy 8-10 darab kis bonyolultságú IC esetében már érdemes a PLD alkalmazását megfontolni. Azt, hogy miként működjön a PLD, különböző módokon is megadhatjuk (ezek a logikai kapcsolatok beviteli módjai):

— Logikai szimbólumok kapcsolási rajzával (ún. sémával).

— HDL nyelven leírt összefüggésekkel (HDL = Hardware Description Language = hardverleíró nyelv).

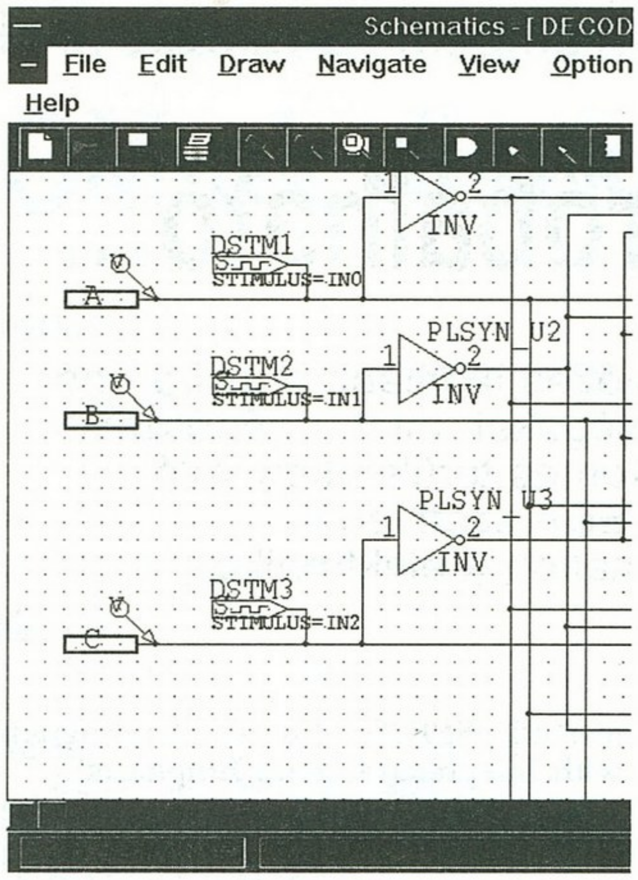
— Már megvalósított kapcsolás ún. Netlist fájljának beolvastatásával (ez a bevált kapcsolás modernizálását teszi lehetővé).

Hogyan válik fizikai valósággá?

A szimbólumok összekapcsolása, azaz a sémaeditálás az első feladat. Új tervezés kezdetén az *és, vagy* kapuk, az inverter, az SR, D vagy J tároló szimbólumát kiválasztjuk az alkatrészkönyvtárból, elhelyezzük a monitor rajzlapján és összehuzalozzuk. Ellátjuk a portkivezetések szimbólumával, és a többit rábízhatjuk a számítógépre, azaz a programra. Ezek a logikai szimbólumok ekkor még ideálisak, késleltetésük nincs, áramot nem fogyasztanak. Hogy elkészült logikai kapcsolásunk működik-e legalább elméletileg, ezt a program szimulációs lehetőségével ellenőrizhetjük.

Az elvi kapcsolás szimulációs vizsgálata következik. A bemenetekre próbaképpen változó vagy konstans logikai szinteket kapcsolunk, és ún. markerekkel figyeljük a kapcsolás viselkedését. Beállítva a szimulációs paramétereket, kiválaszva az ANALYSIS-ben a Simulate menüpontot, a grafikus képernyőn előttünk van a jelek összetartozó csomagja. Ezek tüzetes elemzése után következhet tervünk valóra váltása.

A folytatáshoz először is meg kell adnunk, hogy a könyvtári PLD-készletből melyik IC-családnak milyen toko-



zású, milyen klímaviszonyokra készült változatait vegye figyelembe a program a javasolt eszközlista összeállításakor. TTL, CMOS, ECL, felületre szerelhető, foglalatba kerülő, normál vagy katonai igényeknek is megfelelő változatok közül jelölhetünk. Továbbá el kell döntenünk, hogy számunkra mi a legfontosabb, és mi az, ami fontos ugyan, de nem annyira (vagyis prioritási sorrendet kell felállítanunk az eszközparaméterek között). Beszerzési ár, kivezetések száma, késleltetési idő, maximális frekvencia, felvett tápáram, és még általunk is előírható más követelmények között kell „szavazási" erősorrendet megadni a program megfelelő dialógusablakában.

A beadott megkötéseket, kívánalmakat figyelembe véve, a feladat megvalósítására alkalmas 10 darab PLD főbb adata jelenik meg a gép ajánlatlistáján. Ez a lista önmagában is egy ajánlati sorrend; ezek közül már nekünk kell egyet megjelölnünk. (Az Evaluation változatban persze csak egyféle típust kínál fel a program, a MACH 231-es különféle sebességű változatait. Amennyiben ezzel megelégszünk, átmenetileg még ki is szolgál bennünket ez az olcsó programcsomag.)

Mielőtt bárki lekicsinylené a CD-n közreadott tudásanyagot, el kell mondani, hogy a demó-CD segítségével elkészíthető MACH 231 például 84 kivezetésű, és 128 makrocellát tartalmaz, továbbá újraprogramozhatósága is adott.

A valóságban is jól fog működni?

A kiválasztott típussal újra elvégezve a szimulációt, erre a kérdésre is választ kapunk. Természetesen most már a valóságosat nagyon jól megközelítő, késleltetéseket, terheléseket is figyelembe vevő vizsgálatot végez a programunk.

Jóváhagyott kapcsolásunkról olyan térképet kell készítenünk, amely alapján a PLD programozókészülék a szükséges belső kötéseket el tudja készíteni, be tudja égetni. A mi számítógépünk munkája tehát azzal fejeződik be, hogy a programmal ún. JEDEC fájlt generáltatunk a kapcsolásunk adatai alapján. A programozókészüléket nem is szükséges megvásárolnunk, mert az alkatrészforgalmazók többsége ezt szolgáltatásként nyújtja. Az így megtervezett és „legyártott" áramkör teljesen egyedi, védett az utánépítés veszélyétől, biztonságos a szerzői jogot megsérteni akarókkal szemben.

A kiválasztott PLD eszköz tokozási információit felhasználva netlist fájl készíthető például Pads, Protel, Tango nyomtatott áramköri tervezőprogram számára.

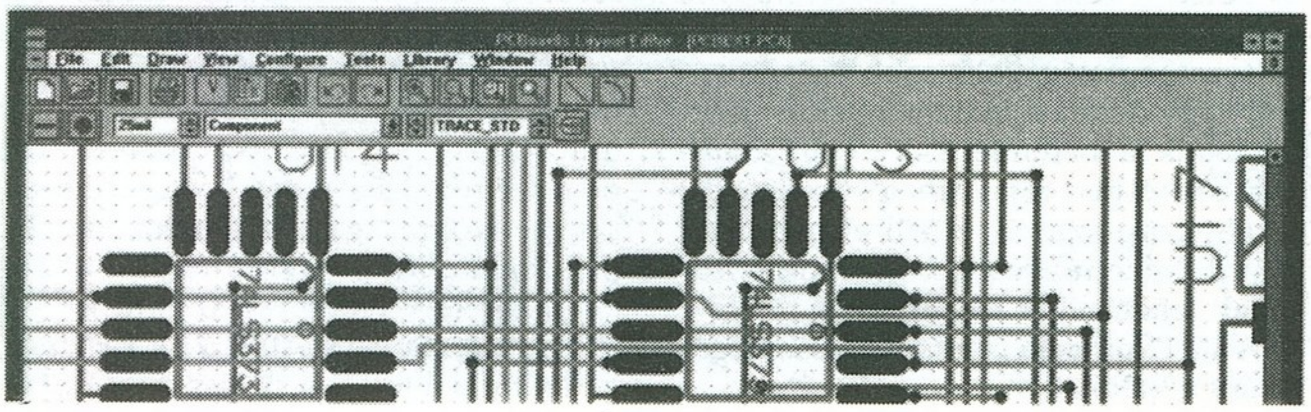
Eddig csak egy, de igen fontos részfeladat megoldásával foglalkoztunk, azzal, hogy a logikai rész megvalósuljon. Viszont áramkörünk környezetében igen gyakran egyéb alkatrészek is szerepelnek. Ezzel a programmal egységes egészként kezelhetjük tervünket, és a nyomtatott áramköri lapon már a teljes megoldás nyer elhelyezést. Például analóg áramköri elemek, csatlakozók stb.

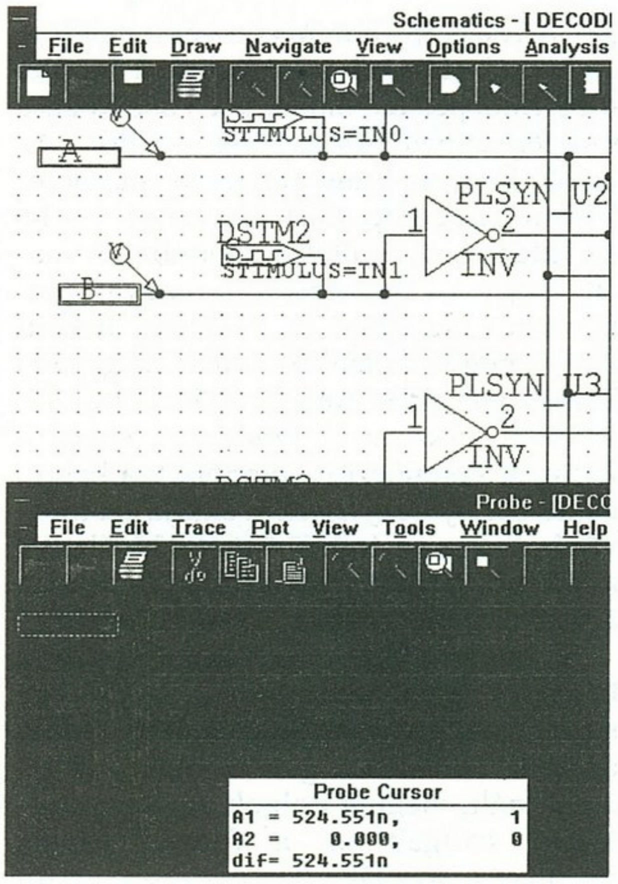
A CD adta lehetőségek

Az elektromos tervezés (is) összetett feladat, ezért a szoftverforgalmazó cégek igyekeznek egy csomagba összegyűjteni a részfeladatok megoldásához szükséges programokat. A MicroSim Corporation Design Center programcsomagja egy univerzális áramkörtervező környezet. Segítségével az áramkörök szinte teljes választéka definiálható és analizálható.

A következő programok demó- és kiértékelő változatai találhatók a CD-n:

— MicroSim Schematics sémaszerkesztő,





— PSpice A/D analóg/digitális szimuláció,

— PSPice Optimizer optimalizáló,

— PLSyn PLD szintetizáló,

— Polaris jeltisztasági modul,

— PCBoards with Autorouter,

valamint a MicroSim tutorial, amely demószerűen elmagyarázza és bemutatja a kérdezett menüpont használatát.

A MicroSim On-line Documentation a CD adta lehetőségeket kihasználva az eredeti dokumentáció több száz oldalas elektronikus „könyv” alakja, fejezetekre tagolva. Benne könnyen lapozgathatunk. Ha a Windows adta párhuzamos futtatást igénybe vesszük, akkor a tervezés folyamán is bármikor „kinyithatjuk”, és utánanézhetünk a kérdéses részeknek. A program szimulációs magját képező PSpice szimulátorhoz a BME-n készített magyar nyelvű távoktatási segédlet is hozzáférhető.

— Hardverigények: 386-os, illetve 486-os gép (32 bites alkalmazási lehetőséggel); 4-8 Mbájt RAM; CD-ROM-meghajtó; lebegőpontos számításokra alkalmas koprocesszor; egér.

— Szoftverigény: Microsoft Windows 3.1 vagy fejlettebb változat.

Szimpatikus vonása a programnak, hogy ha munkánkkal végeztünk, és szükségünk van az általa elfoglalt merevlemez-felületre, az installálás egyik menüpontjaként elindíthatjuk az uninstall futtatást. Ez azután kitakarítja a winchesterről a program által mindenféle könyvtárakba beleírt állományokat.

Az Evaluation programok kipróbálása előtt érdemes a demót megnézni, amely bemutatja a teljes MicroSim programcsomag szolgáltatásait.

Kiss Antal

ARCHICAD

(Vázlat)

— Szöveg kép nélkül, 3 felvonásban —
Történik: XX. század, ezredvég, KÁ-EURÓPA

Játszó személyek:

Új Felhasználó (szövegszerkesztői alapokkal)
Munkatársak (gépileg képzettek és képzetlenek)
Kellékek: Apple Macintosh LC 475, Quadra és ArchiCAD A.55

I. felvonás: Tanfolyam

Helyszín: Módi Stúdió, Nagy Mező.
Szereplők: Reggelenként ajtón be, este mégis haza.
Cselekmény: Játék, fix paraméterek nélkül. Néhány fal a képernyőn: Deim Pál ha látná!
Fáradtsági mutató: Mintha intenzív nyelvtanfolyam.

Utazás a más koponyája körül. Mostantól építész egyformán gondolkodik?
Javasolt alapállás (nevezetesen: úgyis a gépnek van igaza) felvétele.

II. felvonás: „Öldöklő angyal”

Helyszín: Cyberspace.
Szereplők: Három Kezdő Felhasználó + Egy Mester, részleges nyitvatartással.
Időpont: Ősz.
Időszükséglet: Napi 14 óra.

Késleltetett kezdés, többen, több munkán. Raktározni a másik tapasztalatait is. Családi ház, határidő: 1 hónap (kevés lett).
Késsel-villával enni tanulni, felnőttként?
A kézikönyv legfontosabb része csak angolul; a tárgymutató hiánya, összes szocialista papírneműk átka.
Az alapbeállítás csúnyasága.
1:100-ban miért ily részletes rend?
Mégis födémből kéne homlokzatot csinálni, lehet.
A tárgykészítés keserves volta. A negatív lábazat esete a benne lakó ajtóval, szép.
A 3D diszkrét bája.
Megrendelőnek kinyomtatott rajzot nem adunk. Legyen neki színes, szélesvásznú, de nagy képernyőn. Egyébként több ZOOM, mint látvány.

III. felvonás: Kéne már egy gép...

Helyszín: Saját lakosztály.
Szereplő: Félig Kiképzett Felhasználó.
Főszereplők: Diplomamunka, skiccpausz.
Cselekmény: Beszáradt csőtollak szerteszét.

Ó, tűnt ifjúságunk kialakult rajzstílusa!
Meg-fel-be kéne archicadelni a diplomatervet, üdvözlégy, XXI., állandó beruházási kényszer.

Bruckner Csilla

P. S. Szerző számára kölcsöngép felajánlható január, február hónapra. Név és cím a szerkesztőségben.

Három a hetilap!

Közel négy évnyi ingyenesség után januártól a Heti Chip is „megtér" a hagyományos, „fizető" sajtó világába. Akik az elmúlt időszakban megkedvelték, hozzászoktak, bizonyára előfizetés vagy postai árusítás révén is igyekeznek beszerezni a lapot. Ezzel a váltással a Számítástechnika c. hetilapnak a korábbi legfőbb rivális VGA Monitor mellett most már azonos terjesztési kategóriába tartozó harmadik vetélytárssal is meg kell küzdenie az olvasók — és a hirdetők — kegyeiért. (Félig-meddig ebbe a körbe tartozik még a Magyar Hírlap mellékleteként kéthetente napvilágot látó Computer Technika is, amely ugyancsak a hírorientált olvasóréteget célozza meg.)

ElectionLine

Az ABC News, a Washington Post és a Newsweek összefogása révén januártól ElectionLine néven önálló web-oldal készül, amelynek rendeltetése, hogy az amerikai elnökválasztás híreit online módon közvetítse.

Tekintettel az egyidejűleg bejelentkező felhasználók nagy számára, a szolgáltatást a valaha létezett legnagyobb teljesítményű Digital web-szerverre bízzák. Az ElectionLine keretében felhasználják a három cég hírgyűjtési, szerkesztési és riporteri kapacitásait, a 64 bites Alpha architektúra pedig az információáramlás zökkenőmentességét hivatott biztosítani.

(Not) Invalid Directory

A Microsoft Magyarország november 27-én indította könyvtármentő akcióját, amelynek keretében — bevezető Office-kampányának részeként — új pénzügyi forrásokat igyekezett felhajtani az igencsak elszegényedett magyarországi könyvtárhálózat számára. Az akciónak a könyvtárakat érintő legfontosabb eleme a beiratkozók verbuválása, illetve a beiratkozási díjak „felülfizetésére" buzdító propaganda volt. A Microsoft gépezete — úgy tűnik — a felvállalt közérdekű célok érdekében is jól működik, mindenesetre az akció első pár napjában már 20 000-en váltottak olvasójegyet.

A „számmisztikától" vezérelve természetesen mindenütt nyilvánvalóan ott volt a 95-ös szám vagy valamely többszöröse, emlékeztetvén arra, hogy ami a könyvtáraknak jó lehet, az a Microsoft számára sem utolsó lehetőség a publicitásra.

Az akcióban kimondva-kimondatlanul ott lapul viszont az indirekt kritika is: a magyar kultúra ügyéért megint valaki más, egy nem azonos szakmabeli tesz valamit, ha már az állam és annak intézményrendszere erre jelenleg képtelen.

Olimpiát nyerni könnyebb?

Az olimpiák terén továbbra is nagyhatalomnak számít Magyarország. Nemcsak a sportban, hanem a reáltudományok területén is hagyományosan ott vannak az élmezőnyben a magyar fiatalok. Az 1995-ben lebonyolított versenyeken fizikából, matematikából, kémiából, és a lapunkat közelebbről érintő informatikából is olyan eredményeket értek el, amelyek joggal töltenek el bennünket (is) bizakodással. Három — akkor még középiskolás, ma már egyetemista — fiatalember aranyérmet hozott haza a Hollandiában lebonyolított informatikai diákolimpiáról (Kovács Gábor, Radnóti Miklós Gimnázium, Budapest; Blahut György, Szent István Gimnázium, Budapest; Fige Péter, Hermann Ottó Gimnázium, Miskolc). Nekik is, tanáraiknak is gratulálunk, és reménykedünk, hogy mire ők kikerülnek az egyetemekről, már nem külhonban kell érvényesíteniük kivételes képességüket, hanem a magyar gazdaság fejlődése vesz olyan irányt, hogy itthon tudja tartani a tehetségeket.

Huhog a BSA

Amúgy vészmadár módjára huhog. A szoftverek jogi védelmének biztosítására létrehozott szervezet legfrissebb becslései szerint tovább romlott a szoftvermorál Magyarországon, s immár 85%-ra saccolják az illegális szoftverhasználat arányát. A szoftverkalózkodásért — a BSA véleménye szerint — elsősorban a vállalatoknál dívó szoftvermásolási gyakorlat hibáztatható, és a kieső bevételek nagyságát milliárdokra becsülik. A szoftverek kereskedelmi forgalmának érezhető növekedése, a Microsoft, az IBM, a Digital és néhány más cég által az oktatásnak igen jelentős kedvezménnyel juttatott legális szoftverek nagy száma és több más körülmény is sokkal inkább azt a feltevést látszik viszont megerősíteni, hogy nem a BSA által felvázolt tendencia érvényesül. Akik megtehetik, bizonyára szeretnék nyugodtan aludni, míg a továbbra is meglévő, széles körű szoftverbitorlásban sokkal inkább a kemény gazdasági kényszerhelyzet játszik főszerepet, nem pedig az erkölcsi fertő.

Ál-lízing helyett valódi

A Telecomp Pécsről indulva hódította meg a fővárost is, a terjeszkedési stratégia (számítástechnika, irodatechnika, adatátvitel) azonban egy pillanatig sem szorította háttérbe a regionális elkötelezettséget. A Telecomp holding ötéves fennállása alatt első szakmai szimpóziumát tartotta meg Budapesten, ennek két vonatkozása mindenképpen kiemelésre méltó. Az egyik, hogy az Eunet regionális partnereként a Telecomp saját szerverrel lép be az Internet-szolgáltatók táborába, a másik, hogy a cég szeretné a magyar gyakorlatban eluralkodott „ál-lízing" helyett ismét megvalósítani a valódi lízingtevékenységet: előre megállapodott áron vásárolják vissza a lízingidőszak végén a lízingelt berendezést (merthogy a berendezés bérlője *tényleg* nem akarja azt megvásárolni), s az éppen aktuális kínálatból a partner kiválaszthatja azt, amelyet a következő időszakban kíván lízingelni.

S hogy a show is része legyen a kétnapos előadássorozatnak, arról egy speciálisan kedvezményes árakon induló árverési ceremónia gondoskodott. *(Nem* a használt, megunt dolgok elkótyavetyélésére szolgált ez a fórum, hanem vadonatúj, speciális berendezések kerültek kalapács alá.)

Olvasólámpa

A szkenner és az OCR szoftver természetes szövetségesei egymásnak — ezt a felismerést váltja egyre inkább gyakorlattá a két technológia vezető hazai képviselője, a Recognita és a HP. Késő őszi közös akciójuk a technológia-népszerűsítést szolgálta — jelentős sikerrel. Az akciót áthatotta a két cég szakembereinek meggyőződése: az OCR-rel intelligenssé tett lapolvasó épp olyan elengedhetetlen kelléke kell, hogy legyen a számítógépnek, mint amilyen a billentyűzet vagy az egér. A jól kitalált, játékos hangvételű kampány keretében számosan ismerkedhettek meg olyanok is a kézenfekvő alkalmazások lehetőségével, akik eddig csak távolból szerezhettek információkat a rendszerek működéséről.

Üzleti szempontból az is jelentős hír, hogy a kelet-európai képviseletet ellátó HP mellé a tengerentúlon az IBM is csatlakozik: a Recognita Plus OS/2-es változatának Egyesült Államok-beli disztribútora lesz. A Recognita szoftver jelenleg egyébként a 3.0-s verziónál tart, mintegy 140-féle szkennerrel képes együttműködni, egyedülállóan gyors karakterfelismeréssel a világ több mint 80 nyelvén „ért". (Nemcsak géppel írt jeleket, hanem kézzel írt számokat is olvas, sőt a vonalkódok sem hozzák őt zavarba.)

Small Slem

A Compaq december elején azt tartotta a magyar számítástechnikai szaksajtó tájékoztatására legérdemesebb hírének, hogy ismét dollármilliókkal támogat nagy tenisztornákat.

Megrendelés

Floppy Információs Lap

Megrendelem a Floppy Információs Lap című mágneslemez újságot 1 év időtartamra

.... év .... hónapjától kezdődően

.... példányban.

☐ Heti frissítéssel (39.600 Ft + 12% áfa)
☐ Havi frissítéssel (18.920 Ft + 12% áfa)

Az előfizetéshez

☐ Számlát kérek
☐ Átutalási postautalványt kérek

Név: ....

Cím: ....

Helység: ....

Irányítószám: ....

Dátum: ....

/aláírás/

Alaplap Posta

MEGRENDELÉS

Az Új Alaplap 1996/1. számában a 26-27. oldalon ismertetett **szoftverek** közül **postai utánvétellel** megrendelem az alább felsoroltakat:

Név: ....

(Cég:) ....

Cím: ....

Helység: ....

Irányítószám: ....

A szoftverek árát a küldemény átvételekor a kézbesítési díjjal együtt kifizetem.

Dátum:

/aláírás/

Előfizetés az Új Alaplapra

Az 1996/......... számtól kezdődően előfizetem az Új Alaplap c. havi számítástechnikai folyóiratot

...... példányban, ☐ 1 évre, ☐ 1/2 évre.

Az éves előfizetési díj 3564,– forint.

Az előfizetési díj kiegyenlítéséhez:

☐ **Számlát** kérek (banki átutalással fizetek).

☐ **Átutalási postautalványt** kérek.

Név: ....

(Cég:) ....

Cím: ....

Helység: ....

Irányítószám: ....

Dátum: ....

/aláírás/

INFORMÁCIÓKÉRÉS

Kérem, hogy az itt általam **BEKARIKÁZOTT KÓDSZÁMÚ** hirdetésekkel kapcsolatban küldjenek részemre bővebb tájékoztatást.

0101	0111	0121
0102	0112	0122
0103	0113	0123
0104	0114	0124
0105	0115	0125
0106	0116	0126
0107	0117	0127
0108	0118	0128
0109	0119	0129
0110	0120	0130

Beküldhető: 1996. január 31-ig

ÚJ ALAPLAP
1996/1
JANUÁR

FELADÓ **Feladáskor kérjük bérmentesíteni!**

A) Egyéni érdeklődő:

Név: ....

Cím: ....

Helység: ....

Irányítószám: ....

B) Vállalati érdeklődő:

Cég: ....

Ügyintéző: ....

Cím: ....

Helység: ....

Irányítószám: ....

Telefon/Fax: ....

Bélyeg helye

Új Alaplap
szerkesztősége
I., Márvány u. 17.
Pf. 571
Budapest 1539

Belföldön díjmentesen feladható

Új Alaplap
szerkesztősége
I., Márvány u. 17.
Pf. 571
Budapest 1539

Minden PC-hez kell egy jó alaplap!

És egy Új Alaplap!

Belföldön díjmentesen feladható

Új Alaplap
szerkesztősége
I., Márvány u. 17.
Pf. 571
Budapest 1539

Belföldön díjmentesen feladható

Floppy Információs Lap
szerkesztősége

I., Derék u. 6.
Budapest
1016

Az Új Alaplap lemezmellékletének tartalma — 1996 / JANUÁR



Mindent a NetWare-ért

Az utóbbi időben egyre világosabb: a Novell a korábbiakhoz képest jobban koncentrál vezértermékének mind több funkcióval való ellátására. Néhány hetes hír, hogy a Frontier Technologies Corp. bejelentette az első olyan NT-szerver-alapú programterméket, amely a NetWare környezetek számára Internet-elérést, TCP/IP- és host-kapcsolatot nyújt. A Cyber-function nevű szoftver az innovatív átviteli gateway-t a robusztus Internet „host connectivity" alkalmazások egész sorával kombinálja. Ezáltal a NetWare-klienseket az Internethez és más hostgépekhez lehet csatolni, mindennemű sebességi és biztonságtechnikai kompromisszum nélkül, azáltal, hogy nem kell minden egyes desktop gépen TCP/IP-t futtatni.

Időközben a Novell bejelentette, hogy a piacvezető LAN-menedzser programterméke részeként is szállítja a NetWare Navigator 301 verziójú, skálázható vállalati LAN szoftvertelepítő programtermékét. A NetWare Navigator segítségével operációs rendszert, alkalmazást és állományt lehet egyetlen központi helyről telepíteni — és alkalmasint a NetWare LAN-on újabb verzióra cserélni. A Navigator gyorsan és gazdaságosan teríti az adatokat, lehetővé téve, hogy a rendszergazda egy adott központi grafikus felületű rendszeren kövesse a folyamatot.

A Novell 1995 végén összefoglaló útmutatót jelentetett meg a NetWare Telephony Services (NTS) és a Telephony Services API (TSAPI) jövőjéről (új termékekről, technológiákról, programokról). Az NTS-megoldások lehetővé teszik a felhasználók számára, hogy egyszerűen az adathálózatba kötött PC-ről felügyeljék a telefonkészüléket. Az NTS és a TSAPI a piacon az első olyan, hálózat-alapú platform, amely integrálja a számítógépet és a telefóniát (CTI).

Végül e csokorba illik még az a Novell-bejelentés is, amelynek értelmében egy új programtermék, a NetWare TransactionLink révén elérhetők a Tuxedo nyílt rendszerű tranzakciós monitor programcsomag tranzakció-kezelési funkciói, lehetőségei a NetWare 4 felhasználói számára. A NetWare 4 és a Tuxedo integrációja megfelel a Novell egyik stratégiai céljának: osztott szolgáltatások kiterjesztésével és közös alkalmazás-programozási csatolók (API-k) széles körű használatával tenni intelligenssé a LAN-okat.

Alpha gépeken Notes?

A Lotus Development Corp. cég vezető workgroup szoftverét, a Notes-t a DEC Alpha-bázisú rendszerein is meg kívánja jelentetni. Várhatóan 1996 első felében kerül a piacra a Notes for Windows NT nevű programcsomag, majd még a jövő év második felében megjelenik a termék Digital Unixos változata. Stephen O'Neill, a Lotus stratégiai együttműködésekért felelős alelnöke szerint semmiféle teljesítményproblémát nem okoz majd az, hogy a Lotus kifejleszti a 32 bites Notes program első implementációját 64 bites hardverplatformra.

IBM: DB2 adatbáziskezelő Windows NT-n

Az IBM 1995. november vége óta szállítja kliens/szerver technológiájú relációs adatbáziskezelő programcsomagjának, a DB2-nek a Microsoft Windows NT hálózati szerver operációs rendszerváltozatát. Az új DB2 verzió megfelelő szintű megbízhatósági jellemzőkkel rendelkezik, továbbá keresztplatform-adatintegritással és olyan skálázhatósággal, amelyet nem képes tejlesíteni az MS SQL Server az NT-n. Ezt megelőzően a DB2 kliens/szerver változata OS/2, AIX és AP-UX-alapú számítógépeken futott. Tim Negris, az IBM Software Solutions Divisions eladásokért és marketingért felelős alelnöke szerint a DB2-t integrálni fogják az NT-n a Lotus Notes workgroup programcsomaggal és a Lotus SmartSuite desktop alkalmazásokkal.

3Com: integrált távmonitorozás

Az amerikai 3Com olyan eszközkészletet vezetett be teljes termékskálájára, amely egységesíti a monitorozást és a hibafeltárást a cég összes hálózati elemére. A 3Com a Remote Monitoring (RMON) támogatást teljes termékválasztékára nyújtja. Egyúttal a hálózati eszközgyártó megjelent az RMON Client Application termékkel saját, Transcendent nevű hálózatmenedzsment rendszerére. Ez az új Transcendent-alkalmazás az Axon Corp. cég szoftverére épül, amelyet a 3Com menedzsment-platform Unix 3.1-ébe és Windows 4.1 verziójába integrálták.

CA—Unicenter: rugalmasabb, hatékonyabb

A Computer Associates (CA) még 1995 novemberében csökkentette a CA—Unicenter rendszermenedzsment programcsomag árát. Ugyanakkor a szoftver újabban tartalmazza a Legent Corp. cégtől átvett ún. „agent" technológiát is. Az új árstruktúra a tengerentúlon azt eredményezi, hogy a belépési szintű Unicenter ára — kliensenként vagy szerverenként — 700 USD, rendszerenként 400 USD. Az új, csökkentett árak mindazokra a CA—Unicenter temékekre vonatkoznak, amelyek Windows NT, NetWare vagy 500 Unix platformon futnak. A CA időközben bejelentette, hogy a Legent cég Agentworks nevű szoftverének technológiáját a CA-Unicenterbe integrálva hozza forgalomba.

AT&T: újabb integrált multimédia-központ

A múlt év negyedik negyedévében az AT&T Magyarország és az Országos Egészségbiztosítási Pénztár (TB) olyan értelmű szerződést írt alá, amely szerint a TB hang- és adatátviteli integrált intelligens számítógépes és távközlési hálózatát az AT&T fejleszti ki, mégpedig a TB szegedi, szolnoki, tatabányai, szombathelyi és zalaegerszegi megyei igazgatóságai számára. Az AT&T a feladatot mindenütt saját IMX (Integrated Multimedia Exchange) rendszerével valósítja meg. A közel 1 millió dolláros beruházás költségeinek fedezésére a világbanki „Nyugdíjbiztosítási és Egészségbiztosítási Projekt" keretében nyújtott hitel egy részét fordítják. Megyénként az egységes, integrált hálózatot AT&T Systimax kábelezési rendszerrel alakítják ki. A távközlési rendszer a szintén AT&T Definity alközpontokra épül, és kiegészül számlázási rendszerrel, valamint hangposta-szolgáltatással.

Microsoft: Gibraltar a láthatáron

A Microsoft is tervezi kihozni WWW szervertermékét. A bétaváltozatot tartalmazó programcsomag neve Internet Information Server, amelynek korábbi „fedőneve" Gibraltar volt. A termék a WWW oldalak hostjaként működik. A Microsoft Internet Web program — a hírek szerint — nem elégíti ki az üzleti, ügyviteli tranzakciókhoz szükséges biztonságtechnikai követelményeket. A fő cél az volt, hogy egyszerűen lehessen installálni és kezelni.

INFORMÁCIÓKÉRÉS: 0123 ▲

INFORMÁCIÓKÉRÉS: 0124 ▲

INFORMÁCIÓKÉRÉS: 0126 ▲

Minőség és színvonal

Az „optimális" ember–gép kapcsolat

Szakmánk, a számítástechnika, vagy általánosabban az informatika kilépett a „közönséges" tudományok sorából: egyre nagyobb hatású társadalomalakító tényezővé kezd válni. Ha nem akarjuk, hogy ennek a folyamatnak az irányítása kicsússzon a kezünkből, keresnünk kell azokat az eszközöket, amelyekkel a folyamatok kézben tartását meg tudjuk valósítani. A jól érthető közös nyelv is ezek közé az eszközök közé tartozik. Elég néhány alapszempont állandó érvényesítése (olyanoké például, mint igény, funkció, hatás), máris képesek vagyunk a problémák helyes megragadására és lehető legelőnyösebb megoldására.

Tudós futurológusok azt jósolják, hogy az emberiség következő korszaka az informatika kora lesz. Tehát olyan időszak, amelyben — mondják — az előzőktől eltérően az információnak lesz a legmeghatározóbb szerepe. Ez persze elég furcsa jövendölés, mert *amióta a világ világ, azóta ez így van,* csak hát ma már lépten-nyomon információfeldolgozó eszközökbe botlik minden emberfia, a világ leggazdagabb embere is egy szoftvergyáros, így az informatika szerepét már egyes kiemelkedő közgazdászok és társadalomtudósok is kezdik felismerni...

Az információ szerepe sem nem csökken, sem nem növekszik — megmarad, mint mindig is volt, kulcsfontosságúnak. *Az információfeldolgozó technikai eszközök szerepe viszont veszedelmes mértékben növekszik,* és teljesen nyilvánvaló, hogy jövőnk függ attól, milyen lesz az ember és a gép kapcsolata, így elengedhetetlen a tisztánlátás ennek főbb meghatározóira vonatkozóan. A kérdés bonyolult, és sok múlik azon, hogy *sikerül-e megtalálni a lényeges elemek megragadását és kezelését (minél könnyebben) lehetővé tevő eszközöket.*

Igények

Az ember mindig problémahelyzetben van, akár akarja, akár nem, akár tudomást vesz róla, akár nem. Mivel mindennek következményei vannak, mindig valamilyen következményt hozunk létre, és mindig e következmények létrehozása mint feladat megoldását segítjük elő — akár akarjuk, akár nem, akár tudomást veszünk róla, akár nem, akár tudjuk, hogy mik ezek a feladatok, akár nem. Sőt, még az is igaz, hogy minden és mindenki, bármilyen működésével, bármilyen viselkedésével — megint csak akár akarja, akár nem, akár tudomást vesz róla, akár nem — valamilyen igény kielégítésére törekszik, valamilyen igény kielégítésén munkálkodik.

Az igény tehát mindenre vonatkozóan központi fontosságú fogalom. Az embert is mindig igények mozgatják, és egyben korlátozzák. Sokféle, gyakran egymással ellentétes és kielégíthetetlen igényünk is lehet. Az embernek azonban van egy különleges igénye (talán ez az egyetlen igazi igényünk), hogy mindenben, mindenre vonatkozóan a lehető legjobbat szeretnénk. Tevékenységünkben, viselkedésünkben is.

Az igény fogalma, sajnos — különleges hasznossága ellenére — mellőzött, agyonhallgatott fogalom. Ennek oka egyrészt álszemérem. Mintha az igény eleve rossz, „önző" valami lenne; pedig vannak jó igények is. A másik, gyakoribb ok, amikor azért nem beszél valaki az igényekről, nehogy ezzel felhívja a figyelmet saját jogtalan, másokat igazságtalanul hátrányosan érintő igényeire.

Mindenben a lehető legjobbat szeretnénk tenni, a legjobban szeretnénk „működni". Ha tudjuk, hogy mi ez a legjobb, és ezt tesszük, ennek megfelelően viselkedünk, valamint tudjuk azt is, hogy a legjobbat tesszük, és eszerint viselkedünk, akkor a megelégedettség érzése, „csodálatos nyugalom" van bennünk. Ennek a tudatnak a hiánya viszont kellemetlen bizonytalanságérzéssel, kielégítetlenséggel jár.

Hogyan elégíthetjük ki ezt az alapigényünket? Mindenekelőtt az eligazodás hiányát, a tájékozatlanságot, a bizonytalanságot kell megszüntetnünk. Tudnunk kell, hogy vajon jó feladaton (jó igények kielégítésén) és jól dolgozunk-e. Tudnunk kell, hogy mi pillanatnyilag a legfontosabb feladat (igénykielégítési igény), és annak megoldásában hogyan vehetünk részt a leghatékonyabban. *Nyilvánvaló, hogy jó cél csak egy lehet: a lehető legjobb életminőség. Minden feladatnak erre, ennek legjobb elérésére, biztosítására kell irányulnia.*

Mivel pedig nem minden egyforma és egyforma hatású, nem mindegy, hogy mit teszünk, hogyan viselkedünk. Feladataink legjobb megoldásához tehát e tekintetben is nélkülözhetetlen az eligazodás, a tájékozottság, a biztonságos megítélés. (Ez a megítélés, jellemzés lényeges szerepű minőségmeghatározó, és egyben fontos sikertényező is.)

(Meg)ítélések

A helyes minősítéshez, értékeléshez, megítéléshez, jellemzéshez elkerülhetetlenül szükségünk van egy világmodellre. (Ne felejtsük, bevallva vagy elhallgatva, de minden megállapítás egy világmodellre, egy eszmerendszerre és egy filozófiára támaszkodik!) Mivel a világ a lehető legbonyolultabb (mindent tartalmazó) rendszer, azt gondolhatnánk, hogy a világ modelljeinek is szükségképp beláthatatlanul bonyolult alkotásoknak kell lenniük. Meglepő, hogy nem így van. Létezik egy evidencia erejű és rendkívül egyszerű, de általános modell, amely elméleti és gyakorlati szempontból egyaránt jól használható.

A világ semmi más, mint igények rendszere, igények érvényesülési törekvései, ezek eredményei, okai, okozatai, következményei (beleértve az igények kielégülését is), és mindezek viszonyai, megállapítható jellegzetességei. Igény

mozgat, alakít, tart fenn mindent, annyira mindent átható és meghatározó, hogy elég mindannak a megítélésére, ami a világban lényeges. Ha tehát az előbb felsoroltakra vonatkozóan tájékozottak vagyunk, akkor el tudunk igazodni a világban, akkor helyesen és jól tudjuk végezni a dolgok és jelenségek megítélését, és feladataink megoldását is, egyébként pedig nem.

Véleményünk szerint nem helyes, sőt *veszélyes a jövőnket döntően meghatározó ember—gép kapcsolat problémakörét a géppel dolgozó ember és a gép kapcsolatára szűkíteni.* Csak a teljes világ megfelelő figyelembevétele teszi lehetővé, hogy valóban értékes és hasznos megállapításokat tehessünk, még a közvetlen ember—gép kapcsolat szűkebb világában is.

Szerep (funkció)

Irányítási feladatunk megoldásához, a dolgok helyes irányítása érdekében tudnunk kell, hogy milyen szerepe van (lehet) az informatikai eszközöknek (természetesen az emberrel együtt) a társadalomban és a társadalomra (az egyes emberről sem megfeledkezve). Át kell világítanunk a társadalmat, hogy milyen az informatikai élete, az informatikai eszközökhöz, ezek működéséhez és produktumaihoz való viszonya. Tulajdonképpen egy kérdőív kérdéseire kell felelnünk, amelyek olyanokról informálnak minket, mint például az alább felsoroltak.

Az informatikai eszközök, ezek produktumai

— ellenőrzése,
— elterjedtsége,
— fejlesztése,
— használata,
— használatának irányítása,
— hirdetése,
— gyártása,
— javítása,
— jellemzése, minősítése,
— karbantartása,
— tulajdonlása,
— vásárlása stb.

Az informatikai eszközöknek és ezek produktumainak használata

— automatikus tevékenységekre,
— bűnözésre (pl. csalásra),
— elemzésre, szakértői munkára,
— gyártásra, termelésre,
— információtárolásra,
— informálásra,
— irányításra,
— jellemzésre, minősítésre,
— kutatásra,
— modellezésre,
— oktatásra,
— problémafelismerésre,
— problémakezelésre,
— problémakitűzésre,
— problémamegoldásra,
— számítások elvégzésére,
— szórakoztatásra,
— tervezésre stb.

Az információ

— avulása,
— átalakulása,
— birtoklása,
— elosztása,
— előállítása,
— elpusztulása, elpusztítása,
— elzárása,
— feldolgozása,
— felhasználása,
— hatása,
— jellemzőinek alakulása,
— károsodása, torzulása,
— keletkezése,
— kereskedelme,
— megjelenési formája,
— mozgása, áramlása,
— tárolása,
— torzítása,
— manipulálása,
— hamisítása stb.

Az olvasó természetesen folytathatja és finomíthatja a kérdezést. (Ki, mikor, mire használja a gépet? Kire, mire, milyen használat, milyen hatással lehet, van? Hogyan kezeli a társadalom az informatikával kapcsolatos problémákat? Milyen az informálás minősége a társadalomban? És így tovább. Sok érdekes és fontos kérdés tehető fel még például a számítástechnikai munkával kapcsolatban is. Többek között: hogyan végzik a feladatmegoldás, a reprezentáció, a tárolás, az ellenőrzés, a kutatás stb. jellegzetes tevékenységeit?)

Minek mi a hatása?

Ahhoz, hogy az életminőség szempontjából fontos tényezők megnyugtatóan alakuljanak, cselekednünk is kell, használnunk kell a rendelkezésre álló eszközöket. A leghelyesebb viselkedés kiválasztásához a jelenségek és akciók hatásait megbízhatóan meg kell tudnunk ítélni, értékelni kell tudnunk — ezt a megállapítást nem lehet elégszer ismételni. És ez a dolgunk sem megy modellek nélkül, még egészen egyszerű rendszerek esetében sem.

Ahhoz például, hogy megítéljük, kiszámítsuk, mennyivel lesz jobb az informatika korában élni Kelet-Közép-Európában, mint ma, nyilvánvalóan *jellemzőkre, jellemzők értékeire és számolási algoritmusra van szükségünk.* Ahhoz pedig, hogy jó minőségűen sikerüljön kiszámítani, mennyivel lesz jobb az informatika korában élni, ugyancsak *jó minőségű jellemzők (jellemzőértékek), továbbá jó minőségű számolási algoritmus, és ennek jó minőségű végrehajtása kell.*

Ha igaz az, hogy rövidesen belépünk (csúszunk, sodródunk vagy zuhanunk) az informatika korszakába — mellesleg helyesebb lenne, ha informatikai gépek korszakát mondanánk —, akkor az életminőség döntő meghatározója a legtágabb értelemben vett ember—gép kapcsolat lesz. Az életminőség jellemzéséhez tehát *nélkülözhetetlen lenne, hogy az ember—gép kapcsolatot jó minőségűen jellemezni tudjuk. Erre azonban nem vagyunk képesek.*

Egyszerűen azért nem, mert nem állnak rendelkezésre azok a jellemzők, „mutatók", amelyek ehhez a jellemzéshez szükségesek. Tény, hogy a számítástechnika tudománya, a számára és az egész emberiség számára legfontosabb számítások elvégzésére ma még teljesen felkészületlen, mert még a saját szűkebb világának szereplőit, jelenségeit sem képes valamirevaló egzaktsággal jellemezni. (Tehát nem tudjuk egzaktan jellemezni, hogy mi lenne az ideális ember—gép kapcsolat, és így azt sem tudjuk megmondani, hogy mit és hogyan kellene tenni annak érdekében, hogy a reálisan megvalósítható legjobb kapcsolat megvalósuljon.)

A helyzet veszélyes

Mint minden másnak a jellemzése, az ember—gép kapcsolat jellemzése is csak jellemzők („mutatók") rendszerével történhet. Mindnyájan emlékszünk a háromszögnek arra a sok-sok jellemzőjére (jellemző adatára), amivel a szerkesztési feladatokban dolgunk volt (oldal-, súlyvonal-, magasság-, szögfelező-szakaszok hossza, szögek nagysága, belkör- és külkör-átmérő, kerület, terület, nevezetes pontok stb.). Emlékszünk arra a sok hasznos összefüggésre, kapcsolatra is, amelyek a háromszög különböző „adatai" között fennállnak. (Vagy ha nem is emlékszünk mindre, azt tudjuk, hogy számuk bizony tekintélyes.) Elvileg egy háromszög jellemzőinek száma a végtelenségig szaporítható, és az ezek közötti összefüggések mennyisége is nyilván végtelen.

A jellemzők és összefüggések számának végtelen volta (sőt már a nagy száma is) természetesen veti fel azt a gyakorlati kérdést, hogy mi ezek fontossági sorrendje, például abban az értelemben, hogy melyek azok, amelyek gyakrabban és a fontosabb feladatok megoldásában játszanak szerepet. Álta-

lában: hogyan célszerű egy ismeretkör jellemzőit definiálni, hogy azok — a köztük levő összefüggésekkel együtt — valamilyen értelemben extremális („optimális") rendszert alkossanak. Ez a fontos kérdés nemcsak a háromszög, de más rendszer esetében sincs megoldva, sőt *még a probléma tudatosulásánál sem tartunk.*

A jellemzők és összefüggések háromszögnél tapasztalható gazdagságának nyomát sem találjuk a számítástechnikában, sőt még annak környékén sem. Pedig ez a világ sokkal bonyolultabb, mint egy háromszög, és így a jellemzők és a köztük levő összefüggések világa is kellene, hogy gazdagabb legyen. Hogyan is vetődhetne fel ebben a helyzetben a jellemzők közötti válogatás puszta gondolata, amikor nincs mik között válogatni?

A helyzet veszélyes. Nem tudjuk megmondani, hogy minek milyen hatása, következménye lesz. Tehát az irányítás sincs (így nem is lehet) a kezünkben.

A használat mikéntjei

Az informatikai eszközök használatának egzakt jellemezni tudásáról már volt szó. Mire, hogyan használják, lehetne, kellene használni, mikre használódik? Ezekkel a kérdésekkel az ember—gép viszony kényes pontjaihoz is eljutunk. Mennyire kell az eszközökhöz alkalmazkodni? Mennyire tudjuk ezeket, ezek használatát kézben tartani? „Kölcsönösen előnyös" szolgálat fog majd domináln,i vagy az ember egyre nagyobb kiszolgáltatottsága? Milyen lesz a géppel dolgozók, és a géppel, illetve gépen keresztül „ügykezelt", irányított emberek helyzete? A gép már ma is a hatalomgyakorlás eszköze, és egyre inkább az lesz. *A hatalomgyakorlás gép segítségével, gépen keresztül fog történni. Hogyan...?*

A használatnak ezekhez a végzetesen komoly problémáihoz képest jelentéktelennek tűnő kérdés az informatikai, főleg számítástechnikai eszközöknek a kihasználtsága. Ez is nyugtalanító. Világszerte hatalmas mennyiségű gép nincs kihasználva. Egyik oldalon iszonyatos a pazarlás, értékes lehetőségek veszni hagyása, a másik oldalon meg ott a rengeteg tennivaló, aminek elmaradása végzetes lehet. Rengetegfélére lehetne még használni a gépeket. Például időigényes kutatásokra, a jogi szabályozás kialakítására (rengeteg variáns végigelemzését csak géppel lehet elvégezni), és az embernek sok-sok más területen való tehermentesítésére.

A lehetőségeinkhez képest kevés dolgot gépesítettünk, és nem a fontosság sorrendjében halad a gépesítés, hanem a profitverseny szeszélyei szerint. A minőség általában alacsony, legtöbbször a régi elavult szemléletet is belapátolják a gépbe, rögzítik a CD-ROM-okon, az új lehetőségek pedig kihasználatlanul maradnak. Pedig a gépesítés mikéntje és minősége döntő életminőség-meghatározó lesz.

Eddig főleg a gépesítés problémáiról és veszélyeiről volt szó. Rengeteg előnye is lehet azonban. Valóban egyaránt *lehet áldás és átok is.* Ha például tervszerűen használnánk a gépet pontosabb, jobb döntéselőkészítésre, hatáselemzésekre... Ezekhez modellek, pontos leírások kellenek, pontosan és egyértelműen „meg kell mondani" a gépnek mindent. És e munka során sok olyasmire is fény derül, amit eddig természetesnek vettünk, és egzaktan nem fogalmaztunk meg. A gép számára kénytelenek lennénk ezt elvégezni, ezáltal fejlődne a tudomány. Ez lenne egy igazi tudományos forradalom.

A jövőben egyre nagyobb szerepük lesz a szakértő(i) rendszereknek és a tudásbázisoknak. Fenti fejtegetésünkben az olvasó sok olyan elemet találhat, amelyek úgy is felfoghatók, mint *kérdések egy társadalomtudományi szakértő(i) rendszer problémakatalógusából.* Ez nem véletlen, mert e tanulmányt valóban egy ilyen (szakértői rendszerek tudásbázis-használati algoritmusaira vonatkozó) gépesítési célú kutatás néhány fontosabb kérdése inspirálta. Fontos kérdések, amelyek közül azonban nem mindegyikre tudunk válaszolni. Így a minőséget és a színvonalat és az „optimális" ember—gép kapcsolatot érintő kérdésekre sem. Ha azonban ez az állapot sokáig így marad, azt éppen a minőség és a színvonal, és ember—gép kapcsolaton keresztül pedig az emberi életminőség fogja megsínyleni.

Pogány Csaba

„Párhuzamos indokok"

Az egyre nagyobb szoftverrendszerek által kikényszerített hardvernövekedésnek is megvannak a korlátai: a szoftver méretének kiterjedésekor például hatványozottan szaporodnak a hibalehetőségek. Ezért fokozódik a jelentőségük a leíró, deklaratív programozási nyelveknek, közöttük a logikai programozásnak.

Másfelől a hardverelemek sebessége is kezdi megközelíteni az abszolút korlátokat. A további gyorsítás egyik alapvető módja több művelet egyidejű elvégzése. A párhuzamosítás nagyon sokféle szinten valósítható meg: az egy processzoron belüli különféle funkciók párhuzamosításától kezdve egészen a több kontinensre kiterjedő számítógépes hálózatok igénybevételéig. A két véglet között helyezkednek el az ún. multiprocesszoros rendszerek, amelyek több központi egységet tartalmazó, lazább (például üzenetközvetítésen alapuló) vagy szorosabb (például közös memóriát használó) együttműködésre épített rendszerek. Az utóbbi években a multiprocesszoros rendszerek „feljönnek", ma már nemcsak a munkaállomások, hanem a személyi számítógépek gyártói is kínálnak például közös memóriájú, többprocesszoros gépeket.

A multiprocesszoros rendszerek kihasználásának legegyszerűbb módja, ha a különböző processzorokat független feladatok egyidejű elvégzésére használjuk (például multitasking operációs rendszerekben). De mit tehetünk akkor, ha a rendelkezésre álló számítási kapacitást egyetlen feladat minél gyorsabb megoldására akarjuk felhasználni? Ekkor a feladat megoldási algoritmusát kell párhuzamosítanunk, azaz több, egymással valamilyen módon összefüggő részfeladatra bontanunk. Bár a hagyományos programozási nyelvekhez már többféle párhuzamosítást segítő eszköz is kifejlődött, a párhuzamos programok készítése különleges feladat: ui. több, időnként kommunikáló utasításfolyam működését kell a programozónak átlátnia. Ráadásul a kommunikáció erősen időfüggő, azaz két programfutás szinte biztosan különböző időzítést, és így különböző végrehajtási utat eredményez.

Természetesen adódik a gondolat, hogy a párhuzamosítást a programozási nyelv megvalósítása nyújtsa, és így a programozónak ne kelljen ezzel a kérdéssel foglalkoznia. Az algoritmikus programozási nyelvek automatikus párhuzamosítása azonban rendkívül nehéz, mivel ezek a nyelvek a hagyományos, „felülírható" változófogalomra épülnek. Egyáltalán nem mindegy, hogy két utasítást miképp hajtunk végre (az $x=x+1;\ x=2*x$ utasításpár egészen más eredményt ad, ha a végrehajtás sorrendjét megcseréljük). E nyelvekkel szemben a deklaratív programozási nyelvek változói csak egyszer kaphatnak értéket (single assignment principle), és így e nyelvek sokkal könnyebben párhuzamosíthatók.

A reális megvalósíthatóság szempontjából kiemelkedik a logikai programozás közelítésmódja, amely különösen gazdag lehetőségeket ad a párhuzamos végrehajtásra, és ugyanakkor a Prolog nyelvben testet öltve komoly gyakorlati alkalmazhatósággal is bír. Ezt az ígéretes kutatási területet, a párhuzamos logikai programozást tekintjük át.

Vagy- és és-, független és függő

Párhuzamos logikai programozás

Mi a jelentősége a párhuzamos logikai programozásnak, és mi ennek a módszernek a lényege? Milyen szerepet játszanak ezekben a kutatásokban a magyar szakemberek? Miért nem halad gyorsabban a japánok ötödik generációs projektje? Egyebek közt ezekre a kérdésekre is választ ad a szerző az alábbi írásban.

Idézzük fel a logikai programozás (röviden LP) és a Prolog nyelv néhány fontosabb elemét! A logikai programozás alapelve, hogy programjainkat a (matematikai) logika nyelvén, állítások formájában írjuk meg. Például az alábbi Prolog programrész:

```
nagyszülője(U, N):-
    szülője(U, S), szülője(S, N).
```

a következő állításként értelmezhető: 'U'-nak 'nagyszülője' 'N', ha létezik olyan 'S', amelyre igaz, hogy 'U'-nak 'szülője' 'S', és 'S'-nek 'szülője N'. Ugyanennek ez az állításnak van egy végrehajtást irányító értelmezése is: ha az a feladatunk, hogy eldöntsük, hogy egy 'nagyszülője(...)' állítás igaz-e, akkor ezt visszavezethetjük két megfelelő 'szülője(...)' állítás eldöntésére. Vegyük észre, hogy itt a két részállítás eldöntésének sorrendje az eredmény szempontjából közömbös (ámbár a különböző sorrend különböző hatékonysággal járhat).

Hasonlóképpen a

```
szülője(G, S):-
    apja(G, S).
szülője(G, S):-
    anyja(G, S).
```

állítások így olvashatók ki: 'G'-nek 'szülője S', ha 'G'-nek 'apja S', továbbá 'G'-nek 'szülője S' akkor is, ha 'G'-nek 'anyja S'. Ennek az állításnak a végrehajtási értelmezése a következő: ha valamely 'szülője(...)' állítás igazságáról kell döntenünk, akkor ezt vagy egy 'apja', vagy egy 'anyja' állítás eldöntésére kell visszavezetnünk.

Bár az általános logikai programozás szintjén a sorrend nem meghatározott, egy konkrét LP nyelv általában megad egy pontos végrehajtási algoritmust. A Prolog például mind az „és", mind a „vagy" összekötőkkel képzett állításokat a felírás sorrendjében hajtja végre, és az alternatív ágak bejárására mélységi (depth-first) keresést alkalmaz. Azáltal, hogy a programozó tisztában van a végrehajtási sorrenddel, egy rendkívül fontos vezérlési eszközt kap, amellyel a keresési tér nagyságát, és így a futási időt is befolyásolni tudja.

Prolog programok párhuzamos végrehajtása

Egy Prolog program párhuzamos végrehajtása általában azt jelenti, hogy különböző részfeladatokat különböző feldolgozó egységek (processzorok) egyidejűleg hajtanak végre. Mint azt láttuk, a Prolog végrehajtási mechanizmusa a megoldandó feladat részfeladatokra bontásán alapul, ahol ezek a részfeladatok „vagy", illetve „és" kapcsolatban vannak. Ennek megfelelően a logikai programozásban a párhuzamosság két alapvető fajtáját különböztetjük meg: a vagy-, illetve az és-párhuzamosságot.

Vagy-párhuzamosságról akkor beszélünk, ha a Prolog keresési tér egyes ágainak bejárását különböző processzorokra bízzuk. Például a 'szülője(ubul, S)' hívás végrehajtásakor az egyik processzor végzi az 'anyja(ubul, S)' részfeladat megoldását, míg egy másik processzor az 'apja(ubul, S)' részcél megoldásával foglalkozik. A vagy-párhuzamosság lényegéből fakad, hogy a két részfeladat egymástól teljesen függetlenül oldható meg.

Tekintsünk egy kicsit összetettebb példát. Feladatunk legyen egy repülő-

géppel elérhető üdülési célállomás kiválasztása, azzal a megkötéssel, hogy útközben legfeljebb egy átszállást engedhetünk meg. Van egy adatbázisunk a lehetséges járatokról, például:

```
járat(budapest, velence, ...)
járat(budapest, párizs, ...)
járat(párizs, nizza, ...)
járat(párizs, london, ...)
```

Itt a ...-tal jelölt harmadik argumentum a repülőjárat menetrendi adatait tartalmazza (például mely napokon közlekedik, indulási és érkezési időpontok). Ezen adatbázis alapján a következő egyszerű programot írhatjuk meg:

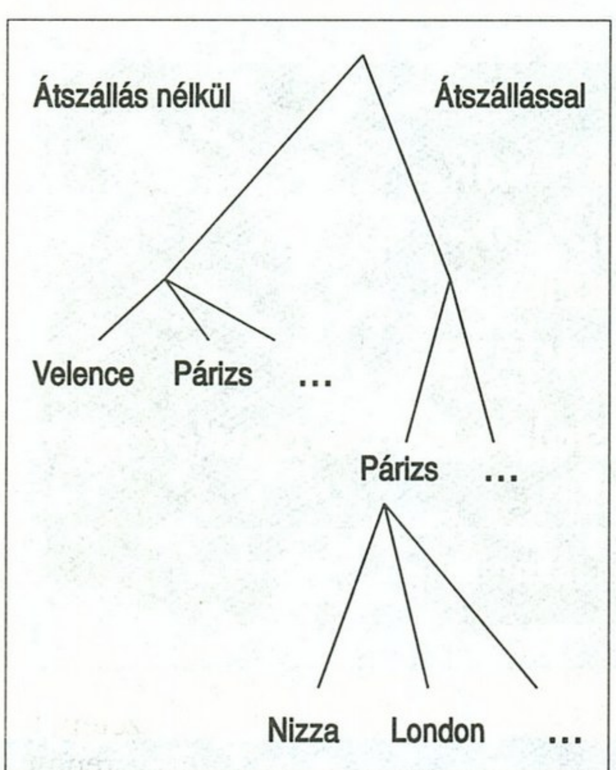
```
célpont(Város):-
    járat(budapest, Város, MAdat)
    megfelel(Város, [MAdat])
célpont(Város):-
    járat(budapest, Átszálló, MAdat1)
    járat(Átszálló, Város, MAdat2)
    megfelel(Város, [MAdat1,MAdat2])
```

A 'megfelel' eljárás az argumentumaiban szereplő adatok alapján dönt arról, hogy az adott célállomás elfogadható-e.

Keresés és megtalálás

A program keresési tere a mellékelt fával ábrázolható. Vagy-párhuzamos végrehajtás esetén a keresési fa minden levélpontjához egy külön processzort rendelhetünk, amely az adott ág feldolgozását végzi (ez a fenti program esetében a 'megfelel' feltétel ellenőrzését jelenti).

Vegyük észre, hogy a vagy-párhuzamosság csak keresési feladatok esetén aknázható ki, azaz csak akkor, ha a programban választási pontok vannak. Az *és-párhuzamosság* ennél általánosabb, mert és-kapcsolatban levő részcélok szinte minden programban előfordulnak.



Az és-párhuzamosság két fajtáját különböztetjük meg, a független és a függő és-párhuzamosságot. *Független és-párhuzamosságról* akkor beszélünk, ha egy feladat két és-kapcsolatban levő részcélja nem tartalmaz (behelyettesítetlen) közös változót. Például tekintsük a mátrix-vektor szorzás alábbi definícióját:

```
mátrix_x_vektor([], _, [])
mátrix_x_vektor([V1|VL], V2, [X|L]):-
    skalárszorzat(V1, V2, X)
    mátrix_x_vektor(VL, V2, L)
```

Itt az első argumentumban a mátrix mint (sor)vektorok listája szerepel. E lista minden elemének és a második argumentumban adott vektornak kell a skalárszorzatát képezni, és az így kapott elemekből képzett lista lesz az eredményvektor. Ennek megfelelően a nem üres mátrix esetét kezelő második állítás a mátrix-vektor szorzás feladatát két független részfeladatra bontja: egy skalárszorzat és egy újabb mátrix-vektor szorzat kiszámítására.

Függő és-párhuzamosságról akkor beszélhetünk, ha két és-kapcsolatban levő részcél egy közös változón keresztül kommunikálhat egymással. Például a korábbi 'nagyszülője' definícióban szereplő két 'szülője' részfeladatban szerepel egy közös 'S' változó. Így a két 'szülője' hívás végrehajtása párhuzamosan indítható, de mihelyst egyikük értéket ad a közös változónak, erről a másikat értesíteni kell.

A függő és-párhuzamosság egy speciális esete az ún. *folyam-párhuzamosság* (stream parallelism). Ebben a közös változó egy lista, amelynek segítségével egy üzenetközvetítő csatorna valósul meg. A lista elemei az egyes üzenetek, amelyek tetszőleges Prolog kifejezések lehetnek. Tekintsünk egy egyszerű programsémát ennek illusztrálására:

```
rendszer(Erőforrások):-
    gyártó(Erőforrások, Termékek), fogyasztó(Termékek)
gyártó(Erő, [Termék|Többi]):-
    legyárt(Erő, Termék, MaradékErő)
    gyártó(MaradékErő, Többi)
gyártó([])
fogyasztó([Termék|Többi]):-
    fogyaszt(Termék), fogyasztó(Többi)
fogyasztó([])
```

Itt a 'rendszer' állításban a 'gyártó' és 'fogyasztó' részcélok egymással párhuzamosan hajthatók végre. Amint a 'gyártó' előállítja a 'Termékek' lista első elemét, a 'fogyasztó' végrehajtása is elindulhat, és az elsőként előállított 'Termék' fogyasztása párhuzamosan folyhat a második termék előállításával, és így tovább. Mint látjuk, a folyam-párhuzamosságot kihasználó nyelvekben szükség van a „gyártó" és „fogyasztó" jellegű részcélok megkülönböztetésére. Ezek megadása, megvalósítástól függően, vagy a programozó feladata, vagy pedig a rendszer automatikusan állítja elő ezeket az ún. globális programanalízis módszerével.

Párhuzamos megvalósítások

A logikai programozás egy ideális párhuzamos megvalósítása a fent felsorolt összes párhuzamosságfajtát támogatná. Ma még azonban nem létezik ilyen univerzális rendszer, a megvalósítások többsége csak az egyik fajta párhuzamosság kihasználását segíti. A következőkben a szerző által legjobban ismert és legfontosabbnak tartott párhuzamos implementációkat tekintjük át röviden.

A logikai programozás párhuzamos megvalósításának feladata már az 1980-as évek elején a kutatások homlokterébe került. Érdekes módon a folyam-párhuzamosság volt az a terület, amellyel több kutatócsoport is komolyan elkezdett foglalkozni.

A legnagyobb problémát a párhuzamos végrehajtásnak a Prolog keresési funkciójával való összeegyeztetése okozta. Ezen a 80-as évek elején úgy segítettek, hogy a Prolog keresési lehetőségeit leszűkítették, és így jöttek létre az ún. CCL (committed choice — elkötelezett választású) nyelvek. A megszorítás lényege, hogy a Prologban meglevő „nem-tudom nem-determinizmus" (don't know nondeterminism) helyébe az ún. „mindegy nem-determinizmus" (don't care nondeterminism) lépett. Prologban ugyanis a teljes keresési mechanizmus lehetővé teszi, hogy akár több helytelen választást is megmásítva, visszalépéses kereséssel találjuk meg az adott probléma megoldását. A CCL nyelvekben viszont minden állításban szerepelnie kell egy ún. elkötelező műveletnek (commit), amely az adott részcélra vonatkozó többi alternatív állítás használatát letiltja (a Prolog „!" műveletéhez hasonlóan). Így a Prologban meglevő automatikus keresés lehetősége megszűnik, bár ez, bizonyos fokig, explicit programkonstrukciókkal helyettesíthető.

A CCL közelítésmód legfontosabb képviselői: a londoni Imperial College-ben kifejlesztett Parlog nyelv, az izraeli

Weizmann Intézet Concurrent Prolog nyelvcsaládja, végül a japán ötödik generációs projektben kifejlesztett Guarded Horn Clauses (GHC) alapú nyelvek. Ezek egymástól lényegében csak abban különböznek, hogy miként adható meg bennük az egyes definíciók fogyasztó-gyártó jellege.

„Történetesen"

A japán ötödik generációs projekt komoly eredményeket ért el a CCL nyelvek hatékony párhuzamos megvalósítása terén. Sajnos a nyelvi megszorítások miatt ez a rendszer nem használható olyan széles körben, mint a Prolog — talán ez is hozzájárult ahhoz, hogy az ötödik generációs projekt nem tudta igazán beváltani a hozzá fűzött reményeket.

A párhuzamos logikai programozás egy másik korai irányzata Magyarországon született. Az 1980-as évek második felében Futó Iván kutatócsoportja által létrehozott CS-Prolog (Communicating Sequential Prolog) rendszer több Prolog egyidejű futását engedte meg transzputeres rendszereken. Ezek a párhuzamosan futó Prolog utasításfolyamok megfelelő beépített eljárások segítségével kommunikálhattak egymással. Bár ez a közelítésmód a párhuzamosság explicit kihasználását jelenti, sok tekintetben hasonlóságot mutat az és-párhuzamosságot implicit módon kiaknázó rendszerekkel is.

Az 1980-as évek végére készültek el az első olyan párhuzamos rendszerek, amelyek a teljes Prolog nyelv támogatását tűzték ki célul. Az ún. Gigalips nemzetközi együttműködés keretében angol, amerikai, svéd és magyar kutatók közreműködésével hozták létre az Aurora nevű vagy-párhuzamos rendszert. Ez a SICStus Prolog rendszerre épülő implementáció közös memóriájú multiprocesszoros rendszereken fut. A párhuzamosan végrehajtott alternatív programágak egyidejű tárolására az Aurora rendszer egy különleges adatstruktúrát, ún. *értéktömböt* (binding array) alkalmaz. Az Aurora rendszer bármely SICStus Prologban írt program párhuzamos futtatására képes, és számos alkalmazásban közel lineáris gyorsulást ér el.

Az Aurorához hasonló a Muse vagy-párhuzamos rendszer, amelyet Svédországban fejlesztettek ki. Ez a megvalósítás az értéktömb helyett a teljes Prolog veremterület másolásával oldja meg az alternatív programágak egyidejű tárolásának kérdését.

Spanyol kutatók fejlesztették ki az \&-Prolog (ejtsd: és-Prolog) rendszert, amely a független és-párhuzamosság kihasználására képes, és szintén közös memóriájú gépeken fut. Ez a közelítésmód fordítási időben kísérli meg felderíteni a potenciálisan független részcélokat (komoly programanalizáló módszereket alkalmazva), és ennek megfelelő feltételes kódot generál.

Számos kutatócsoport foglalkozott azzal is, hogy többféle párhuzamosságot is támogató rendszereket fejlesszen. A független és-párhuzamosság és a vagy-párhuzamosság együttes támogatását biztosította az ECRC (European Computer Research Center) PEPSys rendszere. Az angliai Bristol egyetemén fejlesztették ki az Andorra-I rendszert, amely az Aurorára épülve a vagy-, valamint a függő és-párhuzamosság együttes kihasználását tette lehetővé. Az Andorra közelítésmód egy másik implementációja, az AKL (Andorra Kernel Language) Svédországban készült el.

Számos csoport foglalkozott azzal is, hogy az elkészült párhuzamos Prolog rendszerek használhatóságát vizsgálja. Néhány területen kísérleti alkalmazásokra került sor, és készültek általánosabb hatásvizsgálatok is.

Magyarországon szintén több kutatócsoport foglalkozik párhuzamos logikai programozással. Már említettük, hogy Futó Iván csoportja (ML Consulting) a CS-Prolog nyelvet és transputeres rendszereken való párhuzamos megvalósítását fejlesztette ki. A KFKI MSZKI párhuzamos számításokkal foglalkozó kutatói Kacsuk Péter vezetésével egy adatvezérelt, dataflow elveken alapuló párhuzamos Prolog-megvalósítás kidolgozásán tevékenykedik. Az IQSoft logikai programozási kutatócsoportja pedig a már említett Gigalips nemzetközi együttműködés keretében az Aurora továbbfejlesztésével, illetve kísérleti alkalmazásainak fejlesztésével foglalkozik, például a molekuláris biológia, valamint a bizonytalan tudásbázisok kezelésének területén.

Szeredi Péter

A téma részletei iránt érdeklődőknek a magyar nyelven is hozzáférhető Prolog irodalmat ajánljuk: Farkas Zsuzsa, Futó Iván, Langer Tamás, Szeredi Péter: Az MProlog programozási nyelv — Műszaki Könyvkiadó, 1989; valamint Márkusz Zsuzsa: Prologban programozni könnyű — Novotrade, 1988.

Két új érv az Új Alaplap előfizetése mellett...

1. Az előfizetők ingyen kapják a CD-mellékleteket

Az Új Alaplap első alkalommal az 1995. decemberi számához adott extra CD-ROM mellékletet, a Novellel együttműködve. A későbbiekben hasonló akciókat szeretnénk legalább évente 1-2 alkalommal szervezni. Az állandó floppymelléklet mellett külön CD-ROM mellékletet is tartalmazó számok az újságárusoknál valamivel többe kerülnek (első alkalommal 100 forint volt a felár), az előfizetők viszont ingyen kapják meg a többszáz Mbájtnyi értékes anyaggal megrakott korongot.

2. Gyűrhetetlenné tettük a floppyt

Félévtizednyi küzdelem eredményeként a postai kézbesítők túlnyomó többsége az utóbbi időben már nem gyömöszölte be lapunkat kettéhajtva a postaládába. De azért mindig akadtak néhányan, akik a hajtogatásban hajthatatlanok maradtak. Nekik ettől a számtól kezdve nehéz dolguk lesz, ha a 3,5"-es lemezen ugyanolyan erőszakot próbálnak elkövetni, mint az 5,25"-esen. (A lemezváltás a postaládákat fosztogató és összezúzó vandálok ellen persze nem nyújt védelemet, ezért aki ilyen veszélynek kitett környéken lakik, az inkább a munkahelyi címére fizesse elő az Új Alaplapot.)

...és további 4 klasszikus indok

3. Minden számhoz garantáltan hozzájut

Lapunk nem tartozik a papírhulladékba kerülő kiadványok közé. Nagyon sokan minden számát elrakják, az esetleg hiányzó példányokat pedig évekre visszamenőleg igyekeznek pótolni. Az előfizetés garantálja a hiánytalan sorozatokat.

4. Tíz szám áráért kap 12 számot

Ez 20% körüli árkedvezmény az hírlapárusoknál történő vásárláshoz képest, és anyagilag annak ellenére előnyösebb, hogy a jelenleg 3564 forintos évi előfizetési díjat előre kell kifizetni.

5. Nem érintik az évközi árváltozások

1995-ben a papírárak megduplázódása miatt kénytelenek voltunk év közben a lap eladási árát megemelni. Ilyesmi később is bármikor előfordulhat, de aki előfizetett az Új Alaplapra, azt az árváltozás már nem érinti: előfizetése lejártáig plusz befizetés nélkül kapja lapunkat.

6. Kevesebbe kerül a lap terjesztése

Sokan azt gondolják, hogy a terjesztési költség „a kiadó belügye", amihez az olvasónak semmi köze. Pedig nincs igazuk! Hírlapárusi terjesztésben a lap árának átlag 39 százalékát teszi ki a terjesztői jutalék, vagyis a 356 forintos eladási árból (az áfa befizetését is figyelembe véve) a kiadó mindössze 190 forintot kap! Az előfizetésben kiküldött lapokat ezzel szemben jóval alacsonyabb postaköltség terheli. Ha tehát a terjesztésben nő az előfizetett lapok aránya, és csökken az utcai terjesztésé, az javítja a lap költséggazdálkodását, ezáltal kisebb mértékben emelkedik a lap ára. Az olvasó tehát jobban jár, ha az Új Alaplapnak több az előfizetője!

Ha meggyőztük, hogy 1996-ra érdemes az Új Alaplapot személyesen előfizetnie vagy cégénél megrendelnie, írjon (Pf. 571, Bp. 1539), telefonáljon (156-3211/214-es mellék) vagy faxoljon (201-es mellék), és igényeinek megfelelően mi csekket vagy számlát (vagy mindkettőt) küldünk.

Napirenden: ismét az órarendkészítés

Hogy jól induljon a félév!

Kevés cikk váltott ki akkora érdeklődést, mint az Új Alaplap 1994. februári számában az órarendkészítéssel foglalkozó írás, amelyben Görög András három órarendkészítő programot hasonlított össze egymással. A megjelenés után a szerkesztőséget is, közvetlenül a szerzőt is nagyon sok érdeklődő hívta fel telefonon e témával kapcsolatban. Nemrégiben pedig a PC Szoftver Kft tartott bemutatót a „tökéletes órarendkészítő programról", haladva a korral, most már windowsos kezelői felülettel. A fejleményeket tehát sokan várják érdeklődéssel...

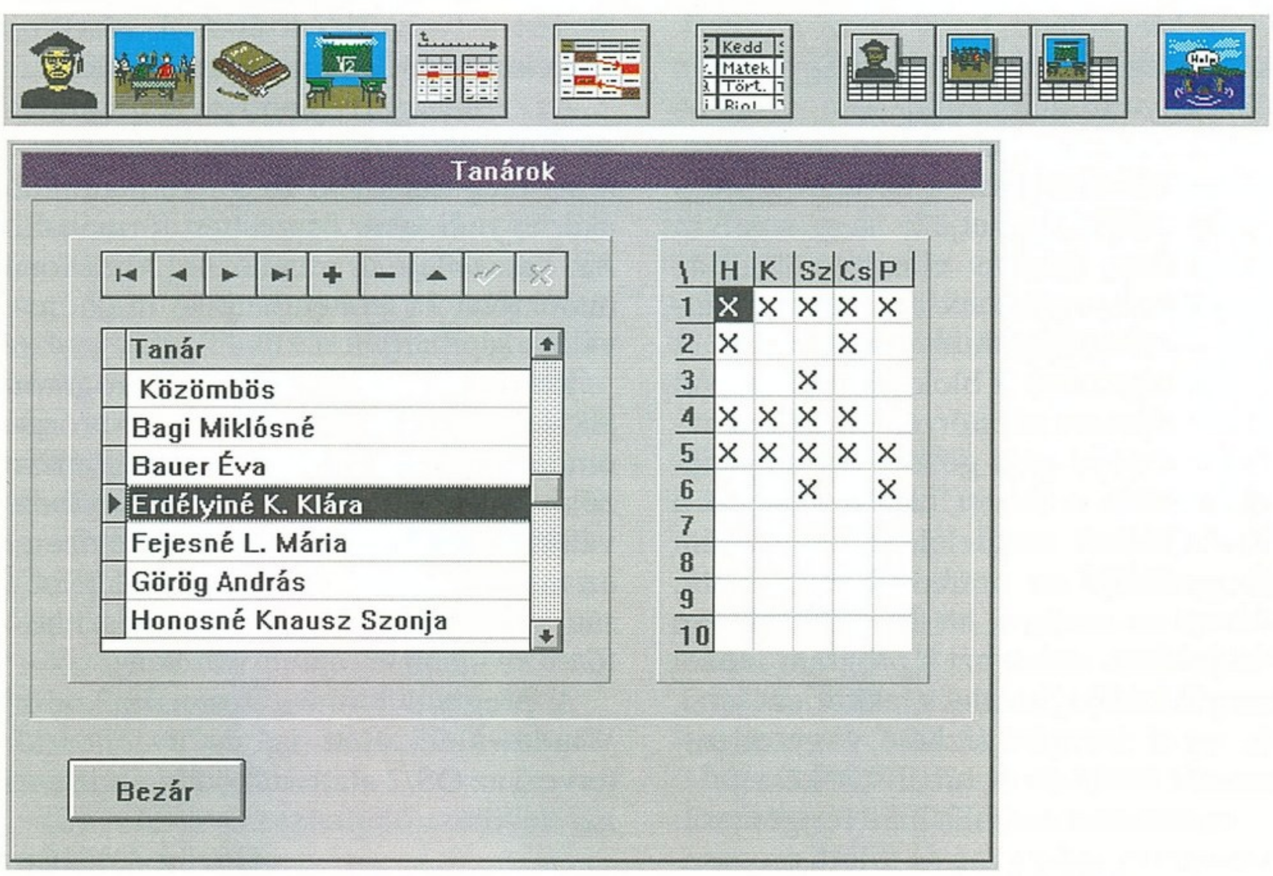
Az 1994. februári cikk óta eltelt időben bőven volt lehetőségem az ismertetett programokat részletesen kipróbálni — több gyakorlati feladat során is. E próbálkozások után arra a következtetésre jutottam, hogy legjobban a Naszádi Gábor által készített program használható. Nem biztos, hogy mindenki olvasta az előző cikket, vagy járatos a számítógépes órarendkészítésben, ezért az alábbiakban bizonyos alapkérdésekre is kitérek.

Részekből az egész

Az órarendek az utóbbi időkben az oktatás nagymértékű specializációja miatt „elvadultak". Hol vannak már azok az idők, amikor egy osztály megbonthatatlan tanulóközösséget is jelentett! Főleg a különféle idegen nyelvek oktatása, nemegyszer többféle szinten, valamint a fakultáció csoportokra tördelte az osztályokat. A csoportok bizonyos tantárgyakra szerveződnek, majd megint másokra átalakulnak. Tehát ma már nem úgy kell feltenni a kérdést, mint hajdan, hogy melyik osztály, melyik tanárral, melyik teremben, melyik tantárgyat tanulja, hanem hogy mely osztályokból létrejött csoportoknak, mely tanárokkal, milyen termekben, milyen tantárgyakat kell egy időben tanulniuk.

Aki már látott órarendkészítőket munka közben, az tudhatja, hogy a fent leírtak szerint összetartozó csoportokat, tanárokat stb. jelentő kartonlapocskákat tologatnak az időkoordinátát jelentő táblán. Célszerű tehát az órarendkészítő program alapobjektumainak ezeket az általunk blokkoknak elnevezett adategységeket tekintenünk.

A feladat tehát: az egy blokkban szereplő tanárokat, osztályokat, termeket az köti össze, hogy feltétlenül ugyanahhoz az időponthoz kell tartozniuk. Azért nem érdemes esetleg félcsoportokban megállapítani a tovább már nem bontható „csoportatomokat", mert a gyakorlatban — nem is ritkán! — előfordul olyan eset is, hogy 2 osztály 5 csoportra bontva, esetleg 5-nél több tanárt is foglalkoztatva tanul egy időben.



Ez egy MI-feladat

Az előbbi gondolatmenetet követve, és a problémát felismerve sokakban megfogalmazódott az a kérdés, hogy nem lehetne-e intelligens programokkal megkönnyíteni az iskolai élet „központi ütemezőjének", az órarendnek az elkészítését. A feladat kezdetben egyszerűnek tűnik: fogalmazzuk meg a tanárok, tantárgyak stb. igényeit valamilyen jól „digitalizálható" feltételrendszerben, és ezután valamilyen hatékony algoritmussal keressük ki a lehetséges megoldások közül a legmegfelelőbbet. Ám aki már maga is megpróbálkozott ilyenfajta, mesterséges intelligenciával bíró program fabrikálásával, az tudja, hogy a kezdeti lendület hamarosan hatalmasan tornyosuló nehézségekbe ütközik.

A szóban forgó programot két darab 3,5"-es floppyn vásárolhatjuk meg a kézikönyvvel együtt. Az installálás kb. 5 percet vesz igénybe, ekkor a program az adatállományokkal együtt a merevlemezre kerül. A programot megvásárlásakor minden felhasználó részére egyedileg regisztrálják, de ettől eltekintve más védelmet a program nem tartalmaz.

A megoldás

A program felépítése is az előzőekben már említett kartonlapocskák tologatásához hasonlít.

Első lépésben a következő adatokat kell gépre vinnünk: tanárok névsora; osztályok (tanulóközösségek); termek listája; valamint a tantárgyak felsorolása. Mind a négy esetben hasonló adatbeviteli panelen kell dolgoznunk. Már az adatbevitel során megadhatjuk, hogy mely tanárnak, osztálynak mikor lehet és mikor nem lehet órája, az egyes tantárgyakat a hét, illetve a nap mely részén kívánjuk majd a programmal elhelyeztetni.

A program újdonsága, hogy a terem mérete szerint megkülönböztethetünk kis és nagy termeket, amelyek a csoportbontásoknál lehetnek fontosak. Meghatározhatjuk azt is, hogy a terem bontóterem lehet vagy sem, azaz más órákat is tarthatnak-e benne egyidejűleg. Ha nem jelöljük ki bontóteremnek, akkor csak a fixen ebbe a terembe kijelölt órákat fogja ide elhelyezni a program. Ha az adatfelvitel során valamit elfelejtettünk vagy módosítani szeretnénk, akkor az a későbbiekben bármikor egyszerűen megtehető.

Ha az alapadatok felvitelével végeztünk, kezdődhet a „kartonlapocskák", az óráblokkok készítése. Egy óráblokk a következő adatokat tartalmazza:

— Tanár (tanárok) neve.
— Osztály (osztályok).
— Tantárgy (tantárgyak).
— Terem (termek), ha nincs konkrét teremigényünk, akkor ?-et írhatunk, és a program majd a szabad bontótermek egyikét foglalja le.
— Blokk mérete (egymás után hány órát tartunk).

A blokk kitöltése után meg kell adnunk, hogy hetente hány ilyen blokkot kívánunk elhelyezni.

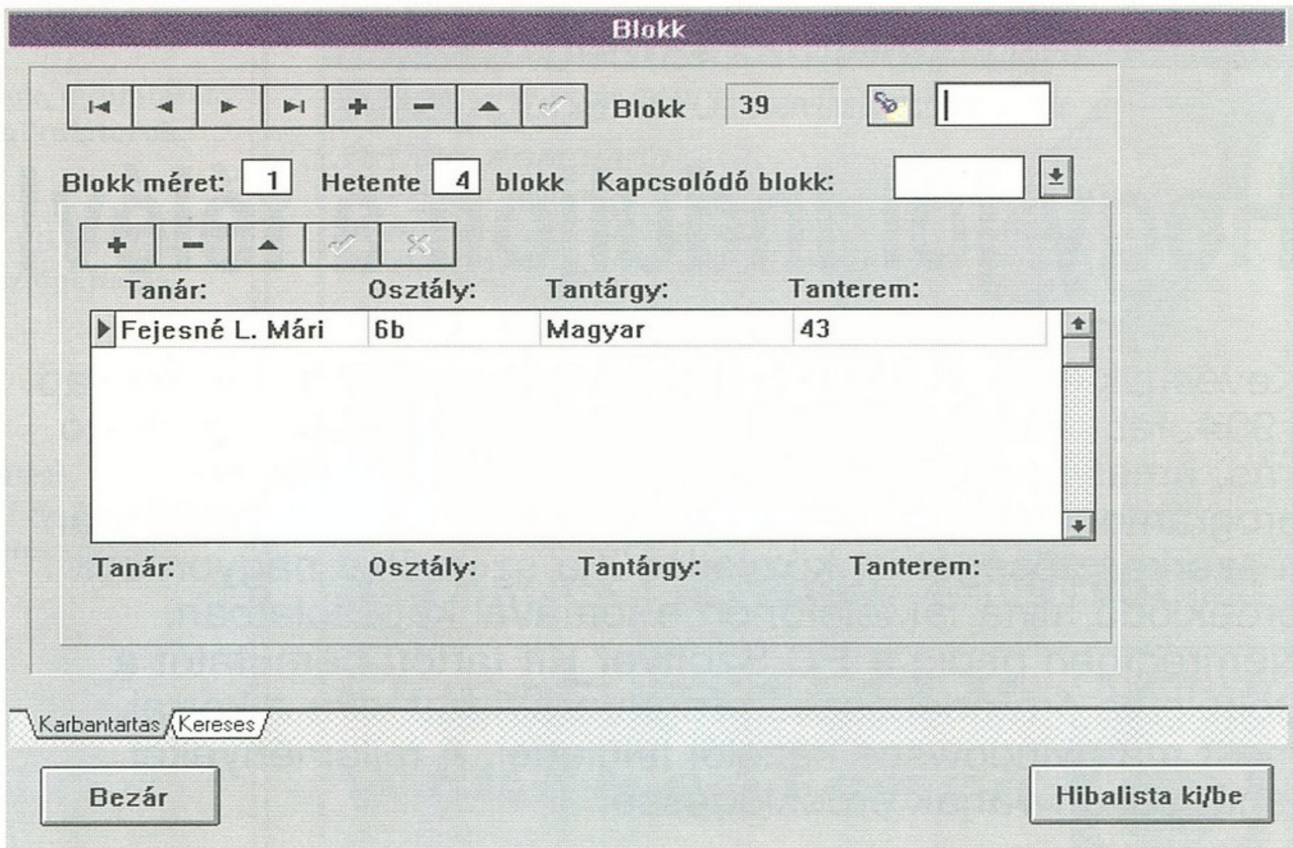
Néhány példa

1. A 7. a osztálynak hetente 3 történelemórája van a 45-ös teremben Kis Péter tanárral:

tanár=Kis Péter
osztály=7. a
tantárgy=történelem
terem=45
egyszerre=1
hetente=3

2. A 6. a és 6. b osztálynak egyszerre van testnevelésórája, a fiúknak Nagy Péterrel, a lányoknak Erős Ágnes tanárnővel, és hetente két dupla órát szeretnének tartani:

tanár=Nagy Péter, Erős Ágnes
osztály=6. a, 6. b
tantárgy=testnevelés
terem=torna1, torna2
egyszerre=2
hetente=2



Az előbbi módszert követve a legkülönbözőbb csoportösszevonások és csoportfelbontások egyszerűen megadhatók. A blokkok kitöltésénél az előzően felvett adatok listájáról kell választanunk, így a félregépelésekből adódó hibák elkerülhetők.

A kritikus dolgok

A gyakorló tanárok jól tudják, hogy megfelelő órarend úgy születhet, ha bizonyos „életszerű" összeférhetetlenségek az órarend készítése során kiküszöbölődnek. Ilyenekre gondoljunk:

— Azonos nevű tantárgyakat egy osztálynak egy napra nem tesz (csak ha dupla óraként adjuk meg), még akkor sem, ha a felvitel külön blokkban történt.

— Megadható, hogy az azonos órák között legyen-e szabadnap, vagy szabad őket egymás utáni napra tenni (csak ha háromnál kevesebb óra van egy héten).

— A blokkok kitöltése után a programmal ellenőrizhetjük, hogy a felvitt adatokban nincs-e ellentmondás, az óraszámok egyeznek-e a tantárgyfelosztásban szereplő óraszámokkal.

Ha végeztünk a blokkok felvitelével és az előzetes ellenőrzéssel, következhet a órarend gépi generálása. A generálás során a felvitt blokkoknak és a feltételeknek megfelelően a program megpróbálja az órarendet elkészíteni. Abban az esetben, ha a feladatnak van megoldása, akkor azt a program idővel meg is találja, ha nincs, akkor ezt jelzi, és az el nem helyezhető vagy ellentmondó blokkokról hibalistát készít.

Itt érdemes szólni a hardverigényről: a program 386-os vagy e fölötti processzort, 4 MB memóriát, VGA monitort és a merevlemezen kb. 5 MB szabad helyet igényel.

Minél fejlettebb processzorunk és minél nagyobb memóriánk van, a program természetesen annál gyorsabb (486 és 8 MB ajánlott), ami elsősorban az adatfelvitelt gyorsítja meg a megjelenítés sebessége miatt. Egy 486 DX2-es gépen egy átlagos órarendet a feltételektől függően 2-25 perc alatt generál a program.

Finomítások

Első lépésben csak a legfontosabb feltételeket érdemes megadni, majd ha így elkészíthető az órarend, akkor lehet tovább finomítani, további feltételek megadásával (például: minél kevesebb legyen a lyukasóra a tanárok részére, helyettesítések előre tervezése stb.).

Az elkészült órarend különböző szempontok szerint (tanár, osztály, terem) megtekinthető és kinyomtatható, akár egyedi, akár összesített formában. Az összesített órarendeket Excel formátumban is elmenthetjük, majd tovább szépíthetjük a külalakját.

Röviden összefoglalva: a program alkalmas szinte minden iskolatípus órarendjének elkészítésére a legkülönbözőbb feltételek alapján. Amennyiben változások állnak be az iskola életében, az eredeti adatok módosításával gyorsan, sok munkát megtakarítva készíthető el az újabb és újabb változat.

A program természetesen működik Windows 95 alatt is, és a fejlesztő tervezi az OS/2 alatt működő változatot is.

Görög András

Tovább bogozzuk az évszázad rejtélyét

Az A oldal labirintusa

A phaisztoszi korong kalandos történetének megismerése (1995/9.) után elkezdtük kibogozni a rejtélyt (1995/12.), amely mindeddig kifogott a kutatókon. Ezúttal egy másik irányból indulunk el a kutatásban: azt vizsgáljuk, hogy mit lehet kihámozni magából a korong szövegéből...

Egy hatvan szó körüli sima prózai szövegben nem valószínű, hogy különösebben nagy szerepük lenne az ismétléseknek. Egy olyan szöveg, amelyben viszonylag sok ismétlés fordul elő, inkább a tudatos költői alkotásokhoz áll közelebb, vagy pedig a korai kultikus, rituális szövegek közé sorolható. Ezek, mint tudjuk, gyakran éppen varázserejük fokozása érdekében tartalmaznak költői elemeket.

Próza — vagy valami több?

Már egy felszínes vizsgálatból is kitűnik, hogy a phaisztoszi korong szövege nem közönséges próza. Ez a tény nagymértékben megkönnyíti a korong szövegének szerkezeti elemzését, hiszen az ismétlődő elemek elhelyezkedését döntő mértékben a szerkezetben elfoglalt szerepük határozza meg. Kölcsönös függésről lévén szó igen sok információt nyerhetünk a szöveg felépítésére vonatkozólag az ismétlődő elemek pozíciójának, jellegének és lehetséges szerepének vizsgálatából.

Elemzésünkben tehát éppen erről az oldalról próbálunk behatolni a korong titkaiba. Vizsgálatunk sokkal több információt próbál meg kisajtolni a szövegből, mint amennyit egy közönséges szövegstatisztikai elemzésből nyerhetnénk, bár a szükséges statisztikai sokaság sem áll rendelkezésünkre. Ahol másként nem tudunk előbbre haladni, élhetünk az emberi intuíció eszközével is, objektivitásunkat azonban minden körülmények között igyekszünk megőrizni.

Vizsgálatunk elsősorban a következő jelenségek megfigyelésére terjed ki:

(1) A szövegben előforduló szavak ismételt előfordulásai.

(2) A kikövetkeztethető szótövek ismétlődései.

(3) A felsorolásszerű (iteratív) elemek előfordulásai.

(4) Egyes szövegrészek egybecsengése, előzőleg hallott szófoszlányok ismételt felidézése.

Mint látható, nem hagyjuk kihasználatlanul azokat az eredményeket sem, amelyeket előzőleg a mikrostatisztikai megfigyelésekből nyerhettünk. Ezek adják az elsődleges információkat ahhoz, hogy plauzibilis következtetéseket vonhassunk le egy-egy szó kisebb részekre tagolásáról.

Ezek alapján valószínűsíthető egyes elemekről, hogy a szóvégi toldalékhoz tartozhatnak, s mindjárt arra is alkalmat adnak, hogy jó közelítéssel meghatározzuk, hol húzódik a határ a szótő és a hozzá járuló toldalék között. (Természetesen maga a toldalék is magában foglalhat szóképző elemet is, a szoros értelemben vett végződésen kívül.)

Tövek és toldalékok

A toldalékolás valószínűsítésének két fő esetével találkozhatunk a korong szövegében:

a) Egyes szótagjelek feltűnően gyakran fordulnak elő a szó végén, vagy feltűnően magas a szóvégi gyakoriságuk a többi pozícióhoz viszonyítva. Ilyen esettel találkoztunk a következő szótagjeleknél:

harcos 19! (100% !)
kutya 8! (73% !)
irha 5! (33%)
támfa 4! (80% !)
sapka 4! (22%)
kés 2 (100% !)
nyíl 2 (50% !)
asszony 2 (50% !)
héja 2 (40% !)
csákány 1 (100% !)
tiara 1 (50% !)

Itt kell megemlítenünk, hogy a „pajzs" (egyébként igen gyakori) jele kizárólag az utolsó előtti pozícióban fordul elő, mégpedig mindig a szóvégi „harcos" jellel összefonódva. Ez az az eset, ahol szinte biztosra vehetjük, hogy egy rendkívül frekventált képzőhöz kapcsolódik a végződés. Nagyon valószínű a képzői szerepe az „irha" jelnek is az A12 és A18 szavakban.

b) A másik eset, ahol a toldalékolás erősen valószínűsíthető, amikor az egyik helyen puszta szótőként fordul elő egy szó, másutt pedig ugyanaz a tő a máshonnan már megismert toldalékkal együtt. Ennek legszembetűnőbb példája mindjárt a szöveg elején látható: az A2 szó toldalékolt változata bukkan fel A6 szóként. Előző listánkból tehát nyugodtan törölhetjük is az „asszony" jelet, amelynek pozicionális gyakorisága új megfigyelésünk értelmében 25%-ra csökken.

Valamivel rafináltabb példa is található az A oldalon: az A24 szó toldalékolt formája lehet a „sarok" + „asszony" szónak, amely feltehetőleg egy összetett szó második elemeként szerepel az A8 szóban. A szótő és a toldalék szétválasztására használt módszerünk így szerencsés esetben összetett szavak felismeréséhez is hozzásegíthet.

Némileg hasonló következtetés vonható le az A31 és a B23 szavak összehasonlításából. Mint látható, mindkét szónak ugyanaz a töve: „sarok" + „lépés" + „jogar". Ekkor viszont a „csákány" jel nem egyedül jelenti a toldalékot, hanem az előtte álló „sapka" jellel együtt.

Ennyi előkészítés után már bátran nekikezdhetünk az A oldal szerkezeti elemzésének. A fontkészlet birtokában könnyen kiteríthetjük a szöveget. Keretes ábránkon a szöveg tördelésében mindjárt az elért eredményeket is érvényesítjük az áttekinthetőség kedvéért.

Szuperhosszú és ultrarövid szavak

A szöveg elején az A1 szó mindjárt felbukkan A4-ként, az A2 szó toldalékos alakja pedig A6-ként. Semmi kétségünk nem lehet afelől, hogy az A4 szóval egy újabb szólam kezdődik, amely az A6 szóig tart, s ez a háromszavas szólam gondolati párhuzama az A1 — A3 szólamnak. Az is valószínűsíthető, hogy a szólamon belül az A2 — A3 egy kétszavas névszócsoport, s a két szólamban az ige ismétlődik meg (A1 és A4). Feltehetőleg a csak szó végén előforduló „nyíl" jel éppen az igei végződést jelöli.

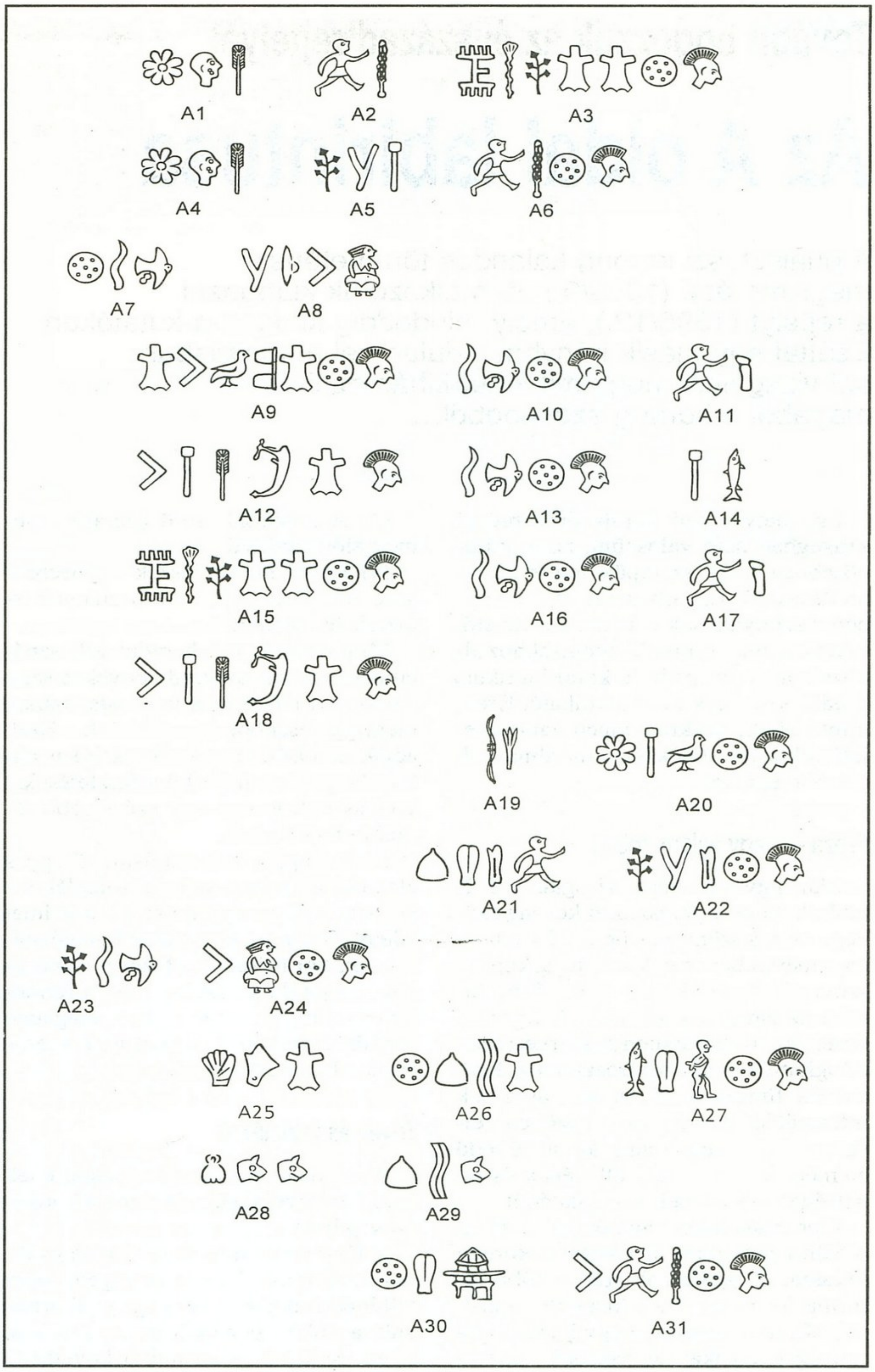
Érdemes felfigyelni az A3 hét szótagos „szuperhosszú" szóra is. Hasonló hosszú szavak fordulnak elő az A9, A12, A15, A18 pozíciókban is, egymástól azonos távolságra, s az A15 szóban éppen az A3 szó ikertestvére bukkan elő. Hasonlóképpen az A18 szóban az A15 szó ismétlődik meg, az A19 szótól kezdve azonban már más a folytatás. A „szuperhosszú" szavakat három esetben ugyanaz a „szarv" + „héja" + „pajzs" + „harcos" szó követi (A10, A13, A16), ezt pedig egy nagyon rövid, két szótagos szó. Az ismétlődés erejét növeli, hogy a rövidke A11 „lépés" + „férfi princípium" szó is ismétlődik az A17-ben.

Szövegünkből tehát kiemelkedik egy ravaszul egymásba fonódó, háromszor három szavas szólamokból álló nagyobb szintaktikai egység, s az A3 szó ennek a nagyobb egységnek a harmadik szólamához, annak első szavához vezet át. A háromszor ismétlődő azonos felépítésű szólam már kezdhetne unalmassá válni. Valószínűleg ezzel magyarázható, hogy negyedszerre is elkezdődik ugyan egy hasonló szólam az A18 szóval (ebben cseng vissza az A12 szó), ennek az új szólamnak a folytatása azonban a „szuperhosszú" szó után megtörik. A töréssel kiemelt pozícióban ismét egy nagyon rövid szócska bukkan fel („íj" + „sáfrány", A18), amelynek a súlyát külön növeli, hogy kezdő szótagjelének, az „íj"-nak ez az egyedüli előfordulása.

Nagyobb egységek is felismerhetők?

A szövegrészek egybecsengésének legszemléletesebb példája az A7 — A8 és A23 — A24 szavak nyilvánvaló hangzásbeli (és minden valószínűség szerint értelmi) párhuzama. A „pajzs" + „szarv" + „héja" (A7) szónak világosan megfelel az „ág" + „szarv" + „héja" (A23) szó, az A8 összetett szó második összetevőjének, a „sarok" + „asszony"-nak pedig az A24 toldalékolt szó töve. Hozzátehetjük, hogy előző megfigyeléseinkkel összhangban az A7 és A23 szavak grammatikai szerepe is valószínűleg azonos: erre következtethetünk a szóvégeken viszonylag gyakori „héja" jelből. Ugyanakkor belső alliteráció is érződhetett az A7 szó második-harmadik jele és a háromszor ismétlődő A10, (A13, A16) szavak között.

Kevéssé valószínű, hogy az A10, A13, A16 szavak tövével egyezne meg az A7 szó második és harmadik jele, s az első szótagot prefixumnak kellene tekintenünk. Ennek a hipotézisnek



ugyanis ellene szól az a kis farkinca, amit eddig nem emlegettünk. Ez a külön berajzolt kisegítő jel teljesen következetesen fordul elő egyes szavak kezdő szótagjánál, a korong mindkét oldalán 10-10 alkalommal. Ez a megkülönböztető jegy konzekvensen ki van rajzolva az A10, A13, A16 szavak tövében, természetes módon hiányzik azonban az A7 és A23 szavakból. Alliterációt tehát joggal feltételezünk, a tövek megegyezését azonban nem.

Gondolati párhuzam

A gondolati párhuzamot az A oldal első és második fele között minden bizonnyal az egymásra „rímelő" A7 — A8, illetve A23 — A24 szópárok biztosítják. Ezek a szólamok csak viszonylagos önállósággal bírnak: minden valószínűség szerint bevezetői egyrészt a háromszor hármas nagyobb szintaktikai egységnek (amely az A18 szótól tovább is folytatódik, egészen az A22-ig), más-

részt a még nem vizsgált A25 — A31 szövegrésznek. Az A19 szóval kezdődő, eddig kihagyott részecske két rövid szólamból áll: az első ezek közül (A19 — A20) olyan felépítésű névszói csoportnak látszik, amilyennel már a szöveg elején találkoztunk (A2 — A3). A második is hasonló, bár lehet, hogy ennek a viszonylag hosszú (négyszótagos) szónak a végén már toldalék is van, s szintaktikai szerepük eltérő. Központi szerepet ebben az A19 — A22 szólamban valószínűleg a töréspont utáni A19 szó játszik, a többi ennek hátravetett kiegészítője lehet.

Az A23 — A24 bevezető szólam után ismét változnak a szerkesztési módszerek. Itt fedezhetjük fel azt a jelenséget, amelyet bevezetőnkben felsorolásszerű (iteratív) elemek előfordulásainak neveztünk. Duplán is előfordul itt ez a jelenség: az A25 és A26 szavakban az „irha" végződéssel, az A28 — A29 szavakban pedig a kutyafejjel. Ez a két-két azonos végződésű szó azonos szintaktikai szerepről árulkodik. Mindkét esetben hátravetett kiegészítő zárja le a szólamot („pajzs" + „harcos" végződéssel), az A30 helyre azonban még egy szó befurakodik.

Hasonlítsuk össze az A30 — A31 szópárt az A5 — A6 szópárral. Azt láthatjuk, hogy az A6 szó jelről jelre megegyezik az oldal utolsó szavának jeleivel („lépés" + „jogar" + „pajzs" + „harcos"). A szöveg elejének az egybecsengése a szöveg végével világosan látható. Az A5 és az A30 szavak között is valószínű azonban, hogy van valami hangzásbeli rokonság: ugyanúgy az „ág" jel felel meg a „pajzs" jelnek, mint az előzőekben vizsgált A7 és A23 szavak esetében, bár most egy farkincával kiegészítve. A hangzásbeli rokonsághoz kevés volna a magánhangzók egybecsengése: sokkal inkább a mássalhangzók egyezésére gondolhatunk.

Bartók találkozása a koronggal

Az egész szövegszerkesztés szinte egy zenemű bujkáló motívumainak elő-előbukkanására emlékeztet. Ennek a megfigyelésnek az alapján az A31 szó szótagtöbblete úgy látszik áthidalhatónak, ha feltételezzük, hogy hangzását tekintve az A31 szó első szótagja, a „sarok" jel az A5 szó utolsó szótagjának, a „bunkó"-nak feleltethető meg, olyan értelemben, ahogy az „ág" a „pajzs"-nak. Ebben az esetben az „ásóbot" jelét az A5 szóban az A30 szó „testiculus" + „házikó" jelpárjával állíthatjuk párhuzamba. Ha e hipotézis beigazolódik, akkor valószínűleg diftongust kell jelölnie e jelpárnak. Tovább pontosítva: az „ásóbot" és a „testiculus" mássalhangzójáról feltehetjük, hogy megegyeznek, a „házikó" pedig egyedül álló magánhangzót jelölhet.

Nem kisebb muzsikus, mint Bartók Béla figyelte meg először, hogy a phaisztoszi korong szövegstruktúrája zeneműre emlékeztet. Hogy mennyire megragadta Bartók figyelmét az éppen akkoriban publikált korong (amelyet utazása során személyesen is láthatott kiállítva Krétán, 1909-ben), mi sem mutatja jobban, mint hogy egy kottát is készített a korong mintájára. Alighanem az ott töltött óráknak — és későbbi feleségéhez, Ziegler Mártához fűződő személyes élményeinek — emlékére készítette Bartók legkülönösebb kottáját. Ezen az ütemek csigavonalban haladnak, mint a korongon a szavak. Bartók kottáját egy amerikai egyetemi közlönyben publikálták, Magyarországra csak egy xerox másolat jutott el az ismert Bartók-kutatóhoz, Demény Jánoshoz.

Hogyan tovább?

Láthatjuk, hogy a belső szerkezet analízise felől tehát szintén elég sok információ kihámozható a korongról. További lényeges előrehaladást természetesen csak attól várhatunk, ha egyesítjük a belső analízisből nyerhető, elég meggyőzőnek tűnő hipotéziseket azokkal a következtetésekkel, amelyeket a lineáris B-vel való formai egyeztetésből nyerhetünk.

Vargha Dénes

Lehet egy programmal kevesebb? Nem!

Az Új Alaplap 1995. decemberi mágneslemezét a meghajtóba téve, és azt a megszokott ALAP paranccsal elindítva ijesztő dolgokat tapasztaltam. (S bizonyára rajtam kívül is néhányan.)

A bevezető zene vége felé a gép hirtelen lefagyott, és fülsértően „üvöltötte" az utolsó hangot. Elhallgattatni csak a Reset gombbal lehetett, a „kéznél lévő" Ctrl-Alt-Del sem segített. Egy darabig makacsul ismételgettem a RESET-ALAP-RESET-ALAP... varázsigéket, amíg fel nem tudtam írni mindent, ami a képernyőn előfordult. Többször is ellenőriztem a gép memóriáját, de a CONFIG és a COMMAND jelenlétén túl az teljesen üres volt, összeférhetetlenség tehát nem merülhetett fel. A sok indítási kísérlet legszebbik „elbocsátó szép üzenete" így hangzott:

```
A:\ALAP [Enter]
MultiKey 2.53 [F.M6SW] betöltve.
(C) Copyright Bata László 1992-94
(C) KeSzo Kft., tel.: 111-8268, 132-8717.
BGI Error: Graphics not initialized (use InitGraph)
A:\NC [Enter]
General Failure error reading drive £
Abort, Retry, Ignore, Fail?
Internal stack overflow
System halted
```

A £ meghajtó olvasásának jogos következménye volt a generálkrach (a General Failure jelentését legjobban talán ez a szó adja vissza, kár, hogy kiejthetetlensége miatt csak nyelvészeti cikkekben szabad leírni), a CONFIG.SYS előírása szerint ugyanis a LASTDRIVE=N volt. Egy PC-guru ebbe biztosan belemagyarázhatna valamit, hiszen az ASCII kódok értéke: N=78, £=156, a fontjel a másiknak éppen kétszerese (egy bit eltolódás!). Egyébként erre a hibaüzenetre válaszolni nem is lehetett, mert az utolsó két sor kiírása is azonnal megtörtént.

A decemberi és a novemberi ALAP.EXE programok belsejében a COMP-pal történő összehasonlítás szerint egyetlen eltérés volt, az ALAP9512 és ALAP9511 alkönyvtárnevek utolsó karaktere, ami logikus. A „BGI Error" üzenetet nem is az ALAP.EXE adja ki, hiszen ezt a program (nem tömörített) szövegében meg lehetett volna találni. Ezzel szemben meghívja az ALAP250.EXE programot, amely (ez esetben sajnálatos módon) a PKLITE-tal tömörítve van, vagyis csak futtatva derül ki, mit csinál. Nos, az ALAP250.EXE zenélt — és fagyott le a „BGI Error" üzenet kiírása után.

Végül a teljes lemezt összehasonlítottam az előző szám lemezmellékletével, és rögtön fény derült a hiba okára. Az én PC-m is, mint másoké, akik már elég régen megvették otthoni munkájukhoz, és nem érezték okvetlenül szükségesnek a cserét, Hercules monitoros. A decemberi szám lemezmellékletéről pedig valahogy lemaradt a HERC.BGI program. Program, hiszen első karakterei szerint „device driver", így a DOS névkonvenciói szerint voltaképpen HERC.SYS-nek kellene hívni, ám a Borland másképp döntött. A HERC.BGI hiányát az ALAP250.EXE-be beszerkesztett grafikus szubrutin felismerte, ki is adta a hibaüzenetet, de vagy magának a szubrutinnak, vagy az azt meghívó programrészletnek a strukturális hibája miatt a veremmutató a továbbiakban nem mutatott a verem tetejére. Ennek következményeit megjósolni nem lehet.

Mivel a decemberi lemezmellékleten 0 bájt szabad hely volt, a számomra érdektelen EGAVGA.BGI-t letöröltem és oda a novemberi HERC.BGI-t bemásolva az én lemezem is futtathatóvá vált.

Szondi Egon János

Könnyebb, mint az egyszeregy?

Az APL jelbeszéde

Napjainkban már mindenféle jelek között élünk. Ha belegondolunk, ésszerűbb is egy piros kerettel ellátott, eldőlni készülő nyilat tartalmazó, háromszögű táblát kitenni, mint kiírni egy nagy táblára: „Vigyázz, veszélyes kanyar balra!” Életünk más területeit is átszövik a jelek; éppen a számítástechnikát, pontosabban a programozást kerülnék el?

A legtöbb és legismertebb programnyelvekben angol szavakat, szótöredékeket használunk az utasítások leírására. Számomra úgy tűnik, hogy az ismét „divatos”, nagyfokú magyarosítás (illetve más nyelvekre fordítás) vakvágány, inkább a Windows által is követett, a Xerox által kifejlesztett ablakos-ikonos jelbeszéd a megoldás. Noha már elkészült olyan programnyelv és programozási környezet, amelyben ikonokból állíthatjuk össze a programot, most mégis egy korosabb, egyszerűbb környezetet mutatok be.

E nyelvben olyan karakterek is szerepelnek, amelyek nem részei a szokványos karakterkészletnek. Talán ez a legnagyobb problémája ennek a programnyelvnek, ami az EGA vagy VGA képernyővel rendelkezők számára nagy gondot nem okoz, mert a karakterek átdefiniálásával leküzdhetik a nehézséget. (Ezt használja ki a lemezmellékleten szereplő program is.)

Az elmúlt évek alatt többen is próbálták valami módon leírni a különleges jeleket, de ezek helyi kezdeményezések maradtak (közülük néhányat bemutatunk a lemezmellékleten is), ám egységes megoldás még mindig nincs. (A nyelv egy olyan továbbfejlesztéséről ejtünk szót hamarosan, amely már csak a szabványos ASCII jelkészletet használja.)

Lássuk a medvét! Melyik ez az alább is sokat dicsért programnyelv? Ez az APL. A több Mbájtra rúgó, Internetről ingyen megszerezhető programcsomagból és dokumentációból kiválogatott minimális, ám még használható rész van a lemezmellékleten. Ez induláskor igen bőséges leírással örvendezteti meg a felhasználót.

„Előítéletek” ellenére

Azok, akik már hallottak valamit erről a nyelvről, és esetleg láttak egy pársoros programot is, most „Na ne!” felkiáltással lapoznának egyet. Valahogy nem tudjuk elítélni őket. Több helyen (így az egykor volt Mikroszámítógép Magazinban is) jelentek meg kisméretű, ám nagy tudású APL programok listái. Ezek alapján szerettük volna megismerni a nyelvet, de a környéken se szakkönyv, se program nem volt. Még ha akadt volna is program, azok a furcsa, ki nem bogozható jelek kínainak tűntek az ember számára. Egy programnyelvet addig úgy tanultunk meg, hogy minimális előismerettel programlistákat bogoztunk ki, és csak ezek után próbálkoztunk saját programokat írni. De hogy lehet egy olyan sort visszafejteni, ahol egyetlen ismerős jel sem akad? (Lásd phaisztoszi korong!)

Szerencsére van megoldás. A C. v. Basum által készített APL2WEB segítségével (a többi Web rendszerhez hasonlóan) egyszerre írhatjuk a program forráskódját a dokumentációjával, és ha elég beszédes a szerző, akkor minden olvasója megérti a programját. Mint ahogy egy Basic program dokumentációjában sem volt az benne, mit is jelent a NEXT szócska, és mire is használható, így egy APL program dokumentációjában se várjuk a speciális jelek elmagyarázását.

A kezdő lökés — ami nélkül nem lehet elindulni — ott van a lemezmellékleten: az IBM által forgalmazott programnak (jelentősen lebutított) demóváltozata. Ha valaki komolyabban szeretne dolgozni ezen a nyelven, vagy nem PC-je van, akkor egy profi változatot érdemes megvennie.

Legfőbb értékek

Nézzük, mire is jó ez a nyelv, érdemes-e egyáltalán betölteni? (Lehet, hogy egy kicsit játszadozni kell, nálam nem indul el minden autoexec.bat és config.sys változat esetén, csakis a Windows számára beállított környezetet szereti.) Mint nagyon sok másik nyelv, ez is általános célú programozási nyelv, tehát amire algoritmus létezik, az APL-ben programozható. De mire érdemes használni?

Főleg számolgatni jó vele, de láttam már APL-ben megírt NetNews böngészőt is. Fő adattípusa a tömb, a gyakorlatban ennek speciális eseteit használjuk, a mátrixokat, vektorokat, numerikus változókat. (A karakterláncok számvektorokként tárolódnak.) Ha valaki olvasott OOP-szakirodalmat, akkor biztosan emlékszik rá, milyen nagy feneket kerítettek annak, hogy a komplex számok összeadását ugyanúgy jelölhetjük, mint az egészekét. Ez itt annyira természetes dolog, hogy senki meg sem említi.

Nem hangzik nagy dolognak, hogy az APL számolásra jó, pedig ez is lényeges alkalmazása a számítógépnek. Bizonyságképp fussunk át néhány lapot, hogy lássuk, mekkora vihart kavarnak az elektronikus számolótáblák (Excel, Lotus) házatáján. Még emlékszem, milyen nagy szó volt a három- és többdimenziós számolótáblák megjelenése, és azok többirányú metszet szerinti vizsgálatának lehetősége. Ekkor az APL-használók magukban mosolyoghattak, mert mindezt az APL már réges-rég tudta, ha nem is olyan látványosan, mint azok, de használhatóbb formában. (Ha belemélyedünk egy számolótábla-programba, láthatjuk: egyre inkább fejlődik az a makrónyelv, amellyel a táblázat feltöltését, használatát automatizálhatjuk. Ha ilyen ütemet tartanak, valaha még elérhetik az APL szintjét is!)

Egyes programozók szeretnek a futtató gép teljes gazdái lenni, elvárják, hogy minden apróság csak az ő engedélyükkel történhessen meg. Az APL

szerénységre inti a programozót, de cserébe el is kényezteti. Itt nem kell semmilyen tárkiosztással és hasonló dolgokkal foglalkozni, ami azzal jár, hogy a program gépfüggetlen lesz. Ez sajnos még nem jelenti azt, hogy ha kiveszem az egyik gépből a lemezt, és bedugom egy másik fajtába, akkor azon is fut a megírt APL program. Ez egy cég különböző gépekre készített programjai között esetleg fennáll, ám amíg nem lesz egységes jelölési mód, csak mindenféle saját készítésű konvertálókkal hozhatjuk-vihetjük programjainkat. Ez kényelmetlen ugyan, de még mindig jobb, mint amikor egy méretesebb C programot az egyik géptípuson lefordíthatunk, míg ugyanaz a másikon hibaüzenettel leáll.

Napjainkban még nincs tisztázva, mit is kellene oktatni számítástechnika címén a középiskolákban. Leggyakrabban a támogató cég által odaajándékozott, vagy valamely tanár által ismert programnyelvet vagy programokat oktatják. A program mellé adott dokumentáció hat tanóra metodikáját is tartalmazza, mely során a diákok (problémamegoldások során) játszva ismerhetik meg ezt a programnyelvet, miközben matematikai ismereteik is bővülnek. Ahogy a program betöltődött, már lehet is játszadozni. Ugyanúgy kezelhetjük a programot, mint egy egyszerű zsebszámológépet, csupán itt nincs felírva a billentyűkre a megfelelő funkció neve. A jelek viszont úgy lettek kialakítva, hogy valamelyest emlékeztessenek a megfelelő funkcióra, erre jó példa az osztás és a szorzás jele.

Különlegességek

Következzen egy-két apróság, amelyre érdemes odafigyelni, mert eltér a megszokottól. A leírásban gyakran szerepel a *monadic* és a *dyadic* szó. Egy műveleti jelet több jelentésben is használhatunk. Vegyük például a kivonás jelét. A nevét onnan kapta, hogy ha elé és mögé is írunk egy-egy számot (a jel mindkét oldalára, tehát innen jön a *dyadic*), akkor az első számból akarjuk kivonni a másodikat. Ha viszont csak mögé írunk (ezért *monadic*) számot, akkor azt negatív előjellel szerepeltetjük. Míg a matematikában csak kevés ilyen többértelmű jel van, az APL-ben kevés kivételtől eltérően az összes ilyen.

A programozást tanulóknak komoly gondot okozhat a hagyományos programnyelveknél használatos precedenciák nem ismerése. Ezek a műveletek egymáshoz viszonyított erősségét, a végrehajtás sorrendjét határozzák meg. Jómagam nem voltam hajlandó megtanulni ezeket a szabályokat, és ezért programjaimban nagy számban (gyakran feleslegesen is) használok zárójeleket. Míg a Pascal vagy C nyelvben körülbelül kilenc csoportba oszthatók a műveletek az erősségük szerint, ez Prolog esetén elérheti a több ezret is. Az APL-ben minden jel azonos erősségű, ám nem a megszokott balról jobbra sorrendben végezzük el a műveleteket, hanem visszafelé. Ezért a *2*3+4*-nek megfelelő APL-kifejezés eredménye nem *10* lesz, hanem *14*.

Gyakran ugyanazt a műveletet akarjuk végrehajtani több számon is. Más programnyelven erre ciklust szervezünk, itt elég leírni a műveletet, és felsorolni a számokat. Ha például a *0.2 0.5 0.7* számokat *15*-tel akarjuk megszorozni, a *15 * 0.2 0.5 0.7* APL-megfelelőjét kell leírni. Ha a művelet mindkét oldalán azonos számú szám áll, akkor a művelet páronként hajtódik végre, azaz *10 20 30 + 3 2 1* eredménye *13 22 31* lesz.

Ha össze akarunk adni tizenöt számot, általában vagy tizennégy összeadásjelet kell írnunk közéjük, vagy egy ciklust kell szerveznünk. Itt a +/ parancsot kell kiadni, és utána felsorolni az összeadandó számokat. A / operátort egyébként tetszőleges más művelettel is használhatjuk, az előtte felsorolt diadikus (kétargumentumú) műveletet hajtja végre az utánuk felsorolt adatokon. Megadhatjuk, hogy a felsorolt számok közül egyszerre hány darabon hajtsa végre a műveletet, így nagyon egyszerűen megkaphatók az egymás mellett álló számok különbségei, szaknyelven fogalmazva az első differenciák. Ha ez idáig még érthető volt, akkor a \ művelet a felsorolt számokból egyre hosszabb sorozatokat készít, és ezeket sorra átadja a / műveletnek, ezért a *+\10 20 30* eredménye *10 30 60* lesz. E speciális operátorok között található olyan is, amellyel a mátrixszorzás is megvalósítható.

Példának okáért...

A mátrixok egyike volt annak a tananyagnak, amelyet középiskolában senki nem tudott a fejembe verni. (Persze ott még nem volt meg ez a program.) Vegyünk példaképp egy boltot, három eladóval (Kovács, Szabó, Zákány), és négy árucikket (A, B, C, D). Minden hónapban készíthető egy ilyen kis táblázat, ahol a kereszteződések az XY eladó által e hónapban eladott Z termékek számát jelentik:

	A	B	C	D
Kovács	3	2	0	1
Szabó	2	2	0	3
Zákány	0	0	1	2

Ha adott a következő táblázat is:

	A	B	C	D
Ár	20	10	12	34
Jutalék	0.2	0.1	0.3	0.1

akkor kiszámíthatjuk (és újfent táblázatba rendezhetjük), hogy ki mennyit árult, és mennyit zsebelt be ezért. Azt a műveletet, ahogy az első két táblázatból az utóbbi megkapható, mátrixszorzásnak nevezzük. (Míg ez a speciális szorzás az APL-nek része, a táblázatkezelő programokban ilyen műveletre nem akadtam, kézzel beírni egy ilyen szorzást viszont embertelenség jókora méretű mátrixoknál.) Az olyan egyszerűbb műveletek, mint a havi eladásokból az éves összeladás kiszámítása, vagy a havi eltérések meghatározása, gyerekjáték.

Aki használni akarja...

A *)Load* paranccsal több példaprogramot is behívhatunk, ezek listájáról a *)Lib* parancs, míg egy példaprogram tartalmáról az *)Nms* ad felvilágosítást, és a programból kilépni az *)Off*paranccsal lehet. Lehetőség van saját programok írására és javítására is, de ez szerintem ebben a demóverzióban (a teljes képernyős editor és debugger hiányában) mazochizmusnak számít. Ha pedig a programhiba helyét akarjuk megkeresni, készüljünk fel a legroszszabbakra. (A „ha már használni akarja, akkor vegye is meg!" eladói filozófia húzódik meg mögötte.)

A példaprogramok közül a legkomolyabbnak a statisztikai csomag (*Stats*) látszik, de ebben csakis picurka programok találhatók. Az APL-re egészében is jellemző a tömörség, a megszokott programnyelveknél gyakran hússzor is rövidebb az azonos eredményt szolgáltató APL program forrása.

Ha valaki többet szeretne APL-ügyben megtudni, akkor a *wuvieai.wu-wien .ac.at/pub/lang/apl* bécsi gépen találhat leírásokat, példaprogramokat. A szerzőkkel az *apl2@vnet.ibm.com* e-mail vagy a *http://www.torolab.ibm.com/ap /apl/apl2.html* WWW-címen találkozhat.

Ne legyünk egyoldalúak: a konkurencia programjáról a *http://www.chilton.com/~jimw/aplnotes.html* WWW-címen olvashat bővebben az érdeklődő.

Bakó Mária

Evidens, hogy az Evolve...

A forráskód TMK-ja

Mindenki „tudhassa": igazán összekutyulni valamit csak számítógéppel lehet. Ez a tény érvényes a szövegszerkesztőkre is. Ki nem járta már meg szövegszerkesztőjének csereműveletével, blokkmozgatásával, eltévesztett mentési felülírásával? Ebben a cikkben ilyen szempontból a szokásostól igencsak eltérő szövegszerkesztőt mutatunk be.

Volt arra eset, hogy adott méretű tömbbel dolgoztam, és e tömb méretével kapcsolatos minden mennyiséget konkrét számmal írtam bele a programba. Egy kis idő múlva persze növelni kellett a tömb méretét, és ez azt is jelentette, hogy újra kellett számolni minden mennyiséget, és mindenütt kézzel átírni az új értékre. (Hogy teljesebb legyen az örömöm, még egyszer megváltoztak a méretek.) Azóta, ha ilyen konkrét értékeket kell használnom, konstansként definiálom, és a vele kapcsolatos számolásokat is inkább a gépre bízom. Ha ezentúl egy ilyen érték megváltozik, elegendő csak egy helyen hozzányúlni a forráshoz, és készen is vagyok. Ilyen könnyebbséget jelenthet számunkra ezután az Evolve használata a programváltoztatásokkor.

Akármilyen pontosan és következetesen is írjuk programjainkat, időnként felmerül a változtatások igénye. Azért, mert ki akarjuk javítani a program hibáit, vagy a megrendelő (esetleg a programozó) újabb funkciókat kíván beépíttetni a programba. Ehhez egyesek a programozói környezetbe beépített szövegszerkesztőket, míg mások kedvenc szövegszerkesztőiket használják. Ezek a szövegszerkesztők többé-kevésbé képesek megtanulni a használt programozási nyelv szerkezeteit, ami megkönnyítheti a forráskód átfutását vagy szerkesztését. Viszont ehhez általában nekünk kell megtanítani ezeket a szerkezeteket a szövegszerkesztőnek, külön-külön minden használt nyelv esetén, ami nem is olyan könnyű feladat, legalábbis nem mindenki számára. (Amikor Pascal programokat írtam, mindig gondot okozott, hogy megtaláljam az adott begin-hez tartozó end-et. Egyes szövegszerkesztők ezt már tudják, de nagy valószínűséggel és egy kis rosszindulattal ezeket a programokat is be lehet csapni.)

Megoldásként

Gyakran nemcsak a szerkezetek felismerésére van szükségünk, hanem mondjuk kíváncsiak vagyunk, hogy egy adott függvényt honnan hívtunk meg, vagy milyen hierarchiájuk van az eljárásoknak. Ezt nagy munkával biztosan meg lehet írni a szövegszerkesztő makrónyelvén is, de vajon megéri-e?

Az ilyen feladatok megoldására remek külső programok vannak, ám ezek statikusak, ami azt jelenti, hogy egy ilyen programot végrehajtva eredményül kapunk egy fájlt, amelyet majd betölthetünk a szövegszerkesztőbe. Ha ezután valamit megváltoztatunk a forrásprogramban, akkor az adott külső programot elvileg újra le kell futtatnunk. Ez a módszer egyrészt megszakítja a szerkesztés folyamát, másrészt sok, számunkra felesleges információt is szolgáltat. (Az ilyen programok ui. egy már kész forrásprogram megismerésében hatékonyak, de egy félkész program írására nem valók.)

Az Evolve lehetővé teszi, hogy menet közben minden extra program nélkül hajthassuk végre ezeket a műveleteket, és ezt tetszőleges programozási nyelv esetén megtehetjük, bár érdemes csak az Algolhoz hasonlító nyelveknél használni (ott, ahol a blokkoké a főszerep).

Bármely nyelven

Minthogy az a cél, hogy a programot arra a programnyelvre is használhassuk majd, amelyiknek az ötlete csak holnapután fogan meg valakinek a fejében, valahogy el kell tudni magyarázni egy-egy adott nyelv szerkezetét. Erre a <^> illetve a :> szolgál majd.

Ha adott egy program forráskódja, akkor könnyű megadni ennek a programnak a hierarchiaszintjeit. Ha egy Pascal programot nézünk, akkor a legfelső szinten a unit-megnevezések, típus- és változódefiníciók, a függvények és eljárások vannak, majd legvégül jön a főprogram. Például a változódefiníciós részt felbonthatjuk a *var* szócskára, és a definíciókat tartalmazó sorokra. Természetesen ezeknek a soroknak is megvan a szerkezetük, elöl egy vagy több változónév, majd egy kettőspont után egy típus neve következik. A program jobb áttekinthetősége miatt szokás ezeket a hierarchiaszinteket megfelelő beljebbkezdéssel (indent) jelölni. Ha a beljebbkezdéseket következetesen használjuk, akkor az Evolve felismeri ezeket a szerkezeteket. Ha csak néhány esetben térünk el ettől, akkor pár billentyűleütés is elég a hierarchiák helyes beállításához.

„Ujjgyakorlatok"

Ha következetesen kihasználjuk szövegszerkesztőnk minden oszlopát, és programunkat a lehető legkevesebb sorba próbáljuk beszorítani, akkor igencsak megizzadunk, mire az Evolve alá igazítjuk a fájlt. Egy kis lazításként javaslom, hogy mindenki kapjon elő egy régi forrásprogramot, és uccu neki, próbálja meg az Evolve szájízére alakítani. (Részfák és sorok mozgatására a hierarchiában a Ctrl és kurzorbillentyűket, illetve Ctrl+PgUp, Ctrl+PgDn, Ctrl+Home, Ctrl+End billentyűkombinációkat használjuk, míg a forrás betöltésére a Ctrl+F8-at.) A már átfutott és helyesnek vélt (vagy adott pillanatban nem lényeges) szerkezeteket eltüntethetjük a Alt+H-val. Ezzel egyszerűbb lesz áttekinteni a maradékot. Ha már nincs szükségünk a „rejtésre", akkor az eltüntetett részeket Alt+F8-cal visszahozhatjuk.

Kis gyakorlás után már mindenkinek a kisujjában van, hogyan mozgathatjuk a programszerkezeteket. Fenti módszert használva nem marad le a feltételes utasítás *else* ága vagy egy *begin* utasítás *end* párja. Ha szükségünk van rá, egy-

szerre több forrásfájllal is foglalkozhatunk, és a vágólap segítségével egész szerkezeteket másolhatunk át vagy mozgathatunk a fájlok között. Természetesen ez is olyan biztonságos, mint a fájlon belüli mozgatás vagy másolás.

Kontrollnak is...

Már ezért is érdemes lehet elővenni ezt a programot, de nem csak erre képes. Megjelölhetünk egy sort, és később ezt a sort átmásolhatjuk az aktuális pozícióba. Ugye, ez eddig még nem igazán érdekes, mert ilyet minden szövegszerkesztő tud. Az viszont már felkeltheti az ember figyelmét, hogy ha az eredeti (megjelölt) sorban valamit változtatunk, akkor ez a változtatás öröklődik az összes másolatban is.

Ez olyankor válik érdekessé, amikor soronként csak egy változónevet szerepeltetünk. Ha programjaink az Evolve szájíze szerint íródtak, akkor a változóütközéseket pillanatok alatt megszüntethetjük. Egy hosszabb program esetén igen nehéz feladat az összes változót külön sorokba pakolni. Ebben is segít az Evolve, a *Marking* menüpont alatti *Search,Extract,Mark/Reference* segítségével egy adott karaktersorozat összes előfordulását külön sorba helyezhetjük.

Az egész program azért született, hogy egy kis programszerkesztés után végül is a fordító számára ehető fájlt kapjunk a Ctrl+F7 paranccsal. Ekkor az Evolve sorra veszi a programunkat tartalmazó fát, és addig írja ki a fában található szöveget, míg nem találkozik egy <^> jellel. Ekkor veszi az aktuális ág alatti alágat, és annak a tartalmát írja ennek a jelnek a helyére. Ha egy sorban több ilyen jel is szerepel, akkor mindegyik helyére behelyettesít egy ilyen sort. Ha a behelyettesítendő sorban szerepel ez a jel, akkor annak a helyére is behelyettesíti annak az ágnak az alágát (és így tovább, rekurzívan). Mindenki próbálja ki, mi történik, ha több (vagy esetleg kevesebb) helyettesítendő sor van, mint hely! A kapott forráskód természetesen tartalmazni fogja a fa hierarchiáját.

Ez eddig mind szép és jó

De ha például ötvenhárom globális változó van, akkor ötvenhárom ilyen jelet kell berakni a fába? Vagy ha később kiderül, hogy kell még két újabb változó, akkor ki kell rakni újabb jeleket? Nem, szerencsére nem kell, mert éppen erre jó a másik speciális jel, a :>. Ezzel a jellel a határoló jeleket lehet megadni. A Pascal esetén az utasításokat, illetve a blokkokat pontosvesszővel, míg a felsorolásokat vesszővel választjuk el. A Pascal program 'main' részét ezért a következőképpen definiálhatjuk: *begin:>;:>end.* , azaz ez a blokk egy *begin* utasítással kezdődik, és majd egy *end* utasítással ér véget, a közte szereplő utasításokat (amelyek ennél a sornál eggyel alacsonyabb hierarchiaszinten vannak) pedig egy pontosvessző fogja elválasztani egymástól.

Az Evolve a közismert Borland-küllakot öltötte magára, ezért bárki képes lehet használni minden tanulás nélkül. A kényelmes munkához viszont meg kell tanulni a speciális billentyűkombinációkat, ami nem megy egyik pillanatról a másikra. A részletes help és a leírás segít megismerni a programot.

Unikum

Ez a program hiányt pótol, semmilyen hasonló képességű programról nincs tudomásunk. Ha valaki előre megfontoltan és mindent megtervezve programoz, akkor az Evolve-ra nem lesz szüksége. Aki kicsit is hebehurgya, ide-oda kapkod, vagy csapatban dolgozik, az valami könnyen elérhető helyre tegye el a programot, mert még jól jöhet a későbbiekben. Programírásra valószínűleg kevesen használják majd, mert a makrók hiányát nehezen pótolhatja az, hogy az ember egyszerre több fájllal is dolgozhat. (Ugyan ki hozna létre egy külön fájlt azért, hogy abban csak egy *if* szerkezet szerepeljen, és azt a vágólap segítségével a programba illeszthesse — a gépelést felgyorsítandó? Ha egy fájlba belerakunk minden programozási szerkezetet, akkor a begépelésnél több időbe telik kikeresni a kellő programszerkezetet, és azt átmásolni.)

Illett volna a speciális kódokhoz is valami billentyűkombinációt rendelni. A program fájának az ábrázolása igen egyszerű, így könnyű gyakorló feladatnak tűnik egy olyan program elkészítése, amely kedvenc programozási nyelvünkön íródott programfájlból olyan Evolve fájlt készít, amelyben minden programozási szerkezet és változó-, illetve konstansnév már külön sorba van szedve. Talán e kis segédprogram segítségével lenne a leghatékonyabb az Evolve, mert így minden programunk azt tehetné, amihez a legjobban ért.

E programot használva sem javul meg „csak úgy" a programozási stílusunk, de legalább megszabadulhatunk néhány apró, de fölöttébb alattomos programhibától.

Aszalós László

Lemezcsere

Lapunk lemezmellékletének történetében 5 és fél év elteltével most került sor a legnagyobb technológiai váltásra: ettől a számtól kezdve az 5,25 collos formátumú lemezről a 3,5 collosra tértünk át.

Decemberi számunk nyomdába adásakor még bizonytalan volt, hogy végül is melyik márka mellett kötünk ki. Amint látható, a KAO győzött. Korábbi kedvező tapasztalatai alapján ez a cég méltányolta leginkább, hogy lemezmellékletünk „udvari szállítójaként" igen jól tudja növelni márkapresztizsét Magyarországon, és a Makrotrend közvetítésével számunkra is megfelelő ajánlatot adott. Lemezeinek kiváló minőségéről a KAO-nak már nem kellett meggyőznie bennünket, de a 3,5"-esek első sokszorosítása a korábbi 5,25"-ös KAO-lemezeknél is jobb eredményt, 1 ezrelék alatti hibaarányt hozott. (Gondolják el, 10 500 másolásból 10-nél kevesebb hibás lemez!)

Nem volt rossz az együttműködésünk 1995-ös szállítóinkkal, a MIC lemezeket importáló Soul Kft-vel és a forgalmazó Teta Kft-vel sem, de a közel azonos összegű új árajánlatok közül most inkább a legnagyobb tekintélyű márkát részesítettük előnyben.

A lemezcserének bizonyos grafikai vonatkozásai is voltak lapunkban, és ez egyúttal más apróbb tartalmi és formai módosításokra is alkalmat teremtett. A figyelmes olvasó bizonyára észreveszi ezeket, de egy kis kommentárt is fűzünk némelyikhez.

— Eddigi arculatának lényeges vonásait megtartva markánsabbá, erőteljesebbé tettük a címlap tipográfiáját, grafikai megoldásait, hogy ne legyen már annyira „Hamupipőke" a hírlapárusok standjain a harsány laptársak társaságában.

— Megváltoztattuk a lap alcímét. Úgy éreztük, hogy pontosabban kifejezi lapunk jellegét a tartósságra, elmélyültségre utaló FOLYÓIRAT megnevezés, mint a csillogást-villogást, a felszínességet sugalló MAGAZIN szó.

— Új rendszerben készítettük el az éves tartalomjegyzéket. Feltüntettük például a szerzők nevét is, hiszen az [Új] Alaplap mindig is számos első publikációnak minősülő, sehol máshol nem olvasható szakmai anyagot tartalmazott, és ezt a profilját tovább fogjuk erősíteni.

— Új rovatokat indítottunk. Az ARGUMENTUM az erőteljesen érvelő, vitatkozó okfejtéseknek, olykor provokatív új gondolatoknak a rovata lesz. A MÉDIA a kommunikációs eszközökkel foglalkozó írásoknak ad helyet. A VIZIT a „gyárlátogatásokon" szerzett információhalmazból közvetíti az olvasó számára általunk leginkább érdekesnek tartottakat.

— Az információkérő kártya beküldési határidejét lerövidítettük, mert 2-3 hét arra elegendő.

— Alkalmanként extra CD-ROM melléklet is lesz lapban, ha megfelelő témára és a költségek egy részét vállaló cégekre bukkanunk.

A fenti változtatásokhoz felhasználtuk az 1995-ös közvéleménykutatásunk során olvasóinktól kapott észrevételeket is. A felmérés eredményeinek bővebb ismertetésére hamarosan sort kerítünk.

Faklen Pál

A NEM KERESKEDELMI CÉLÚ EGYÉNI HIRDETÉSEK KÖZLÉSE INGYENES

A kereskedelmi célú apróhirdetések tarifája gépelt soronként (azaz 60 karakterenként) 300 forint. A terjedelem alapján így kiszámított összeget kérjük az Új Alaplap Kiadói Kft számlájára átutalni (OTP, 11701004–20171649), vagy postautalványon a kiadó címére elküldeni (1539 Budapest, Pf. 571), és feltüntetni, hogy „Új Alaplap, apróhirdetés". A befizetést igazoló szelvény másolatát — a hirdetési szöveggel együtt — a szerkesztőséghez (a kiadóéval azonos címre) küldjék el.

Szerzői jogokat sértő szoftverhirdetéseket nem közlünk le.

Bármilyen típusú szöveg fordítását vállalom angol-magyar, magyar-angol nyelvre, illetve kiadványok látványtervezését, szerkesztését is. Cím: Lachner Zoltán, 1195 Budapest XIX., Jáhn Ferenc u. 14/a. Telefon: 157-0308.

OBJECTS 2.0 — objektumorientált programozás CLIPPER-ben. Tájékoztató kérhető az alábbi címen: Szűcs János, 4400 Nyíregyháza, Vasvári Pál u. 37. Tel.: (42) 437-331 vagy 465-666/1382-es mellék.

Adatmentés CD-re, streamerre; winchesterről, floppyról. Ugyanitt beszerzési tanácsadást, hálózattervezést és programkészítést is vállalok. Cím: Kovács Lajos, 1031 Budapest III., Vízimolnár u. 10. IV/33.

Stúdiómban megbízhatóan, ellenőrzötten lefordítom angol, német, francia és magyar nyelvről/nyelvre műszaki és közgazdasági folyóiratok cikkeit, hardver- és szoftverleírásait. Áfás számlát állítok ki. Cím: Szász György, 1035 Budapest III., Kórház u. 25. Tel.: 168-4874.

Alaplapcsere, memória-, winchester- és floppybővítés a helyszínen. MegaSoft. Telefon: 295-5085.

486DX4 100 MHz-es használt/felújított alaplapot és SoundBlaster hangkártyát **keresek.** Cím: Palotás Ákos, 2071 Páty, Ady E. u. 21.

Megalakult az MPC. Ha van olyan shareware program, amely csak neked van meg, vagy te írtad, vagy szeretnél bármilyen ingyenesen terjeszthető programhoz hozzájutni, írj a következő címre: Hadar Dávid, 8143 Sárszentmihály, Fő út 127.

Eladók régi és új számítástechnikai folyóiratok 150 Ft/db áron. Továbbá eladó 30 db 5,25"-os lemez tele hasznos programokkal 90 Ft/db áron. Ezenkívül még eladó a következő számítógép konfiguráció: AT 286-20 MHz, 40 MB HDD, 1,2 + 1,44 MB FDD, mono monitor, billentyűzet, egér. Ár: 65.000 Ft. Cím: Kocsis Zoltán, 5901 Orosháza, Dózsa Gy. út 13.

Eladó **8748-49 mikrokontroller égetőadapter** + szoftverek a Rádiótechnika '95/10-ben közölt eprométegetőhöz, 2000 Ft-ért. Cím: Magi István, 3532 Miskolc, Gyula út 54. Tel.: (46)337-074.

Levelezőpartnereket, számítástechnikához értő barátokat, barátnőket keresek tapasztalatcsere céljából. Ha lehet, rendelkezz saját PC-vel (ha értesz a .BAT fájlokhoz, keress)!!! Cím: Kovács Gábor, 3502 Miskolc II., Pf. 83.

Digitalizálótábla (tablet), 12x12 colos vadonatúj 20.000 Ft-ért eladó! Ugyanitt Atari Portfólió IBM kompatibilis zsebszámítógép (palmtop) soros interfésszel 22.000 Ft-ért eladó. Telefon: 180-6732.

Számítástechnikai oktatás a legolcsóbban! Részletfizetési lehetőség, Budapesten saját otthonában tanulhat. Vásárlási tanácsadás, vírusvédelem, DOS, Windows, programozás stb. Telefon: (06 30)414-571 vagy 203-9144.

Keresem a CoV 1-10. számait megvételre! Ugyanitt eladók olcsón CoVox (MS Windows-kompatibilis) hangmodulok. Válaszborítékot küldj! Cím: Varga Zoltán, 9731 Kőszeg, Pf. 9.

Notebook 486DX-33, 8 MB RAM, 520 MB winchesterrel eladó. Irányár: 230.000 Ft. Telefon: (22)314-352 vagy (88)425-088 munkaidőben.

Eladó az alábbi számítógép-konfiguráció: PC 386DX (486DX-re! bővítést támogató alaplap), 4 MB RAM, Hercules mono monitor, 1,2 FDD, 40 MB HDD, 101 gombos billentyűzet, 3 gombos egér, monitorszűrő. Ár: 55.000 Ft. Tel.: 118-6066/2635 (az Új Alaplap megjelenését követő héten!)

Disznóölés floppy-lemezen! Több ezer recept, ötlet, módszer leírása IBM-kompatibilis lemezen. Ára: 290 Ft. Megrendelés utánvéttel! Válaszborítékért katalógus. Cím: Bartek és Társa Bt., 8206 Veszprém, Pf. 869.

GYERE ÉS JÁTSSZÁL VFX1 sisakkal a virtuális valóságban! Repülés, stratégia, szimulátorok, Doom-szerű játékok, akár egymás ellen is. Címünk: VR Stúdió Bp. V., Irányi u. 5. I/2. Telefon: 137-1160. Nyitvatartás: K-P: 14-21, Szo: 10-21, V: 14-20. Kedvezmények!

Az Új Alaplap 1995. évi összesített tartalomjegyzéke

A HÓNAP TÉMÁJA

A TÖBBI ROVATBAN

NYÍLT TÉR

(Vitaindítók, esszék, eszmefuttatások)

NYÚZÓPRÓBA

(Tesztelés)

OKTATÁS

(Iskolai, oktatási témák)

PALETTA

(Színes, illusztrált termékhírek)

PRO DOMO

(Szerkesztőségi, kiadói információk)

PROGRAMOZÁSTECHNIKA

(Programozási fogások, eljárások, módszerek, programnyelv-ismertetések)

SZERSZÁMOSLÁDA

(Hasznos hardver- és szoftvereszközök)

SZOFTVERPORTÉKA

(Kereskedelmi forgalmazású szoftverek)

SZÖVEGELŐ

(Nyelvi szoftverek, szövegszerkesztés)

TUDÁSTECHNOLÓGIA

(A mesterséges intelligencia és alkalmazásai)

UNIXUMOK

(A Unix világa)

VISSZACSATOLÁS

(Megjelent anyagokra reagáló cikkek és olvasói észrevételek)

Az Új Alaplap 1995. évi összesített tartalomjegyzéke

A LEMEZMELLÉKLETEN

1995. évi számaink hirdetői

16 színű BMP—>EXE konverter — BMP.TXT, BMPX#.EXE (Simay Endre István)
Memorix címnyilvántartó program — MEMORIX.DOC, MEMO#.EXE (Nagy Sándor)
Adatvédelmi segédprogram — MST.TXT, MST#.EXE (Kiss Péter)
A Dindraw rajzolóprogram — DIND#.EXE (Dinnyés Márton)
A klasszikus Mahjongg egyik windowsos átirata — MAHW#.EXE

AUGUSZTUS

Floppymeghajtók átírása — A_BKONV.EXE, B_AREGN.EXE, BALAP.BAT (Simay Endre István) BATKONV.TXT, SWAPAB.COM ➾ 57. o.
A hónap témájához: alternatív Program Manager, sőt több... — WAYF#.EXE
Az adatbázis-iskola illusztrációi — BESZ#.EXE (Ferenczi Gábor) ➾ 27. o.
Példaprogramok a Fortran-sorozathoz — FOR2#.EXE (Szondi Egon János) ➾ 53. o.
Prolog nyelvjárások és példaprogramok — MIK1#.EXE, MIK2#.EXE, PPID#.EXE ➾ 45. o.
A Crush szuperkompresszor — CRUSH.EXE ➾ 34. o.
Magyar fejlesztésű szövegszerkesztő — BIHA#.EXE (Bihari Zsolt)
Directory-kezelő program — GDIR.DOC, GDIR.EXE (Andreas-Josef von Gencsy)
Dámajáték kínai változatban — CHIN#.EXE

SZEPTEMBER

ComDic számítástechnikai szótár — COMDIC.DOC, COMDIC#.EXE (Péteri László)
Reklámcímcsokor V. — REKCIMEK.TXT
Rendszerátvilágító program — SYSC#.EXE
Példák a Fortran-sorozathoz — FOR3#.EXE (Szondi Egon János) ➾ 54. o.
Assembly példaprogramok — ASSM#.EXE (Pethő Ádám) ➾ 58. o.
Látványos apróságok — FRACTALS.COM, JUSTDOIT.COM
Egy ukrán tömörítő — JAM#.EXE ➾ 49. o.
Floppyleltározó Szlovéniából — FMAS#.EXE ➾ 49. o.
Mit tegyünk, ha az 5,25-ös floppymeghajtó B: elnevezésű? — BATKONV2.TXT
Windowsos helpek „ablaktalanítója" — SMDO#.EXE
Master Mind, számjegyekkel — MASTER.DOC, MASTER.EXE (Bányai Zoltán)

OKTÓBER

Az ALAP maradt, csak a BAT-ból EXE lett — ALAPBAT.TXT ➾ 50. o.
CD-ROM olvasók Novell alatt — CDCMMNDS.TXT, CDNOV#.EXE ➾ 37. o.
Két program Internet-használatra — CFTP#.EXE, TWS#.EXE ➾ 28. o.
Windows-ikonok rajzolása DOS-ban is — ICONEDIT.EXE, ICONEDIT.ICO, FILE_ID.DIZ (Doma Zsolt)
Színkeverő — COLORMIX.TXT, COLORMIX.EXE (Boronkay Gábor—Simay Endre István)
BMP-ből EXE — BMPEXE.TXT, BMPEXE.EXE (Simay Endre István)
Gráfszerkesztő és analizáló program — STEFI#.EXE (Tóth Bálint)
Önkormányzati sorozatindító — ONKORM.TXT, KORVED.TXT (Pogány Csaba)
Egy labirintusváltozat — SLABIRIN.TXT, SLABIRIN.EXE (Simay Endre István)

NOVEMBER

Iratbiztonság, egyedi pecsét — MEZEY.TXT (Mezey Gyula) ➾ 18. o.
Egy jelszóprogram — JELSZO.DOC, JELSZO.EXE, JELBE.EXE (Pál Zsolt) ➾ 12. o.
Prímszámgenerátor és faktorizáló — PRIMES.DOC, PRIMES.EXE (Tóth Bálint) ➾ 4. o.
Naplózóprogram, sőt több — NAPLO#.EXE (Erdélyi Tibor)
A RAR parancsai és paraméterei — RARPAR.TXT (Ford.: Nagy Gábor) ➾ 40. o.
Ami a 95/8. szám lemezmellékletéről lemaradt — UNCRUSH.EXE
A HTML Assistant for Windows hipertextjelölő — HTML#.EXE ➾ 43. o.
A Fifth és az Until — FIFTH#.EXE, UNTIL#.EXE ➾ 54–57. o.
Környezeti hatástanulmány, döntéstámogatás — KVED2.TXT, MIN.EXE (Pogány Csaba)
Az amőbaprogramozási pályázat „demója" — AMOBA.EXE (Csiki András) ➾ 61. o.
Pókerjáték — POKER#.EXE (Dinnyés Márton)

DECEMBER

Címadatkezelési minták — CIMAD#.EXE (Ferenczi Gábor) ➾ 5. o.
E-mail címforrások — ASZAL#.EXE (Válogatta: Aszalós László) ➾ 10. o.
Smart Address címregiszter Windowsra — SM#.EXE ➾ 5. o.
A kreatív levél — KREADM.TXT (Hamburger Béla) ➾ 12. o.
Programozástechnikai készlet — NOWEB#.EXE (Válogatta: Aszalós László) ➾ 55. o.
Phaisztoszi korong TrueType-ban — EV100AD.TXT, PHAISZ#.EXE (Vargha Dénes, Slawomir Zabaglo) ➾ 39. o.
Háztartási készletnyilvántartó — MIRELIT.EXE (Kőkuti Tamás)
Tanulásmódszertani segédlet — TANUL.EXE (Pogány Csaba)
„Egy lapra tett" Rubik-kocka — KOCKA.TXT, KOCKA.EXE (Simay Endre István)
Búcsú az 5,25-östől — BUCSU.TXT

AZ EXTRA CD-ROM MELLÉKLETEN

PerfectOffice 3.0 (90 napos verzió), Novell Messenger, NetWare kliensek, Novell javítások (patch-ek) és demók

A fizika évszázada

Mikroelektronika minden szinten

Könyvismertetéseinkkel most a műszaki megalapozásban szeretnénk segítségére lenni olvasóinknak. Két értékes könyvet ajánlunk a figyelmükbe a Műszaki Könyvkiadó kínálatából. Mindkét könyv elsősorban a gyakorlati problémák között igazít el, olvasásuk közben azonban alkalmunk nyílik mélyebb elméleti problémákkal is megismerkedni.



Dr. Mojzes Imre (főszerk.):

Mikroelektronika és elektronikai technológia

Műszaki Könyvkiadó, 1995
408 oldal, 2490 Ft.

A Műszaki Könyvkiadó és a Budapesti Műszaki Egyetem már régóta készülődött egy nagy összefoglaló mű kiadására, amely a végzett vagy leendő villamosmérnökök számára megkönnyíti a felzárkózást a mikroelektronika területén a nemzetközi színvonalhoz. A régen várt mű most megjelent. A könyv önmagában természetesen csak az alapokat teremtheti meg a tájékozódáshoz — az állandó lépéstartást kizárólag a szakirodalommal való rendszeres foglalkozás, és (nem utolsósorban) a nemzetközi tudományos életbe való aktív bekapcsolódás biztosíthatja.

Egyetlen szerző — még oly sokoldalú tudásanyaggal és tapasztalatokkal rendelkezzék is — nem lenne képes ennek a sokfelé ágazó és hihetetlen sebességgel fejlődő tudományágnak az átfogó ismertetésére. Mojzes Imre műegyetemi professzor következetesen érvényesítette azt az elképzelését, hogy minden területet a szakma leginkább szakavatott ismerője írjon le. A könyv célja így sem lehetett az egyes résztereületek teljes mélységű megismertetése, erre 400 oldal nyilvánvalóan kevés lett volna. A közvetlen cél sokkal inkább az értő egységbe foglalás, valamint a nemzetközi színvonalat biztosító korszerű technológia módszereinek és eszközeinek a megismertetése.

Részletes leírás található a könyvben a félvezető-alapú mikroelektronikai elemek fő gyártástechnológiai műveleteitől és az integrált áramkörök alkatrészkészletétől kezdve a monolit áramkörein, a moduláramkörök szerelőlemezein, szerelőtechnológiáin és számítógéppel segített tervezésén keresztül a diszkrét elemek és az elektronikus részegységek konstrukciójáig számtalan fontos területről.

Az optikai hírközlés fontos területeként tárgyalja a könyv az optoelektronika legújabb eredményeit, elsősorban ehhez kapcsolódva kerül sor a vegyület-félvezetők tárgyalására is. A mikroelektronika határterületeként külön fejezet mutatja be az érzékelők, a mechatronika, a mikromechanika és a különleges bevonatok világát.

Ennek a széles skálájú ismeretanyagnak az adott keretek között lehetséges mélységű ismertetése nemcsak a szorosabb értelemben erre a területre szakosodott szakemberek számára lehet hasznos, hanem azoknak is, akik közvetlenül vagy közvetve kapcsolatba kerülnek a mikroelektronika valamelyik ágával.

Érdemes külön megjegyezni, hogy számos helyen találhatók a könyvben orientáló, értékelő megjegyzések, sőt külön alfejezetek is, amelyek felhívják az olvasó figyelmét a leginkább perspektívikus fejlődési irányokra, módszerekre. A különböző eljárások lényegének bemutatásán kívül a szerzők tudatosan törekedtek arra, hogy az olvasó mérlegelni tudja az alkalmazott módszerek, eljárások előnyeit és hátrányait, rámutatva közben arra is, hogy a legújabb kutatások miképpen próbálják kiküszöbölni vagy legalább csökkenteni a negatív tényezőket.

Egy ilyen sokak számára szóló, de mégis erősen szakmai tematika kifejtése során nem könnyű kialakítani a helyes arányt a mindenki számára érthető szóhasználat és a gyakran szakmai zsargonba hajló szaknyelv között. Az elfogadott magyar és idegen nyelvű szakkifejezések kerülése vétkes könnyelműség volna. Szerencsére a szerzők nem esnek sem a túlzott magyarítás, sem a szakzsargon halmozásának hibájába. Igyekeznek közérthető nyelven megfogalmazni a magyarázatokat és bevezetni a szakmai nyelv elfogadott terminusait. Bizonyára sokan örömmel fogadják, hogy a szakkifejezések mellett mindenütt megtalálhatók a nemzetközi szakirodalomban használt közkeletű elnevezések, rövidítések, méghozzá kiemelt betűtípussal. Ezek megismerése nélkülözhetetlen feltétele az irodalomban való önálló tájékozódásnak, a szakma korszerű művelésének.

A könyvet a Műszaki Könyvkiadó gondozta, a tőle megszokott alapossággal. Külön dicséretet érdemel a könyv kiváló ábraanyaga, amely nagyban elősegíti a könnyűnek éppen nem mondható ismeretanyag elsajátítását. A könyv használatát segíti az elég részletes (bár távolról sem teljes) tárgymutató. El lehetne viszont gondolkozni rajta, hogy a következő kiadásban nem lenne-e érdemes fogalommagyarázatokból is összeállítani egy függeléket azok számára, akik mindennapi szókincsébe nem tartozik bele a getterezés, az epitaxia, a detergens oldat vagy a tixotróp oldószer.

A szilicium karrierje

Aki csak a felhasználás oldaláról közelített eddig a mikroelektronikához, talán el sem hinné, milyen nagy szerepe volt a legutóbbi évtizedek elektronikai fejlődésében a fizika, a kémia és a technika összefogásának. Az elméleti alapok megteremtésében elsősorban a kvantummechanikának volt döntő szerepe, a gyakorlattal való kapcsolat kiépítése azonban már egy új tudományágnak, a szilárdtest-fizikának az érdeme. Ez az újonnan született diszciplína jelentősen felértékelte a műszaki tudományokat, valamint a műszaki technológiát mint a tudományok fejlődésének szerves alkotórészét.

A hetvenes évek elejére a fizikusoknak már legalább a fele a szilárdtest-fizika bonyolult problémáinak a megoldásán munkálkodott. Az elméleti célkitűzés az volt, hogy visszavezessék a szilárdtestek tulajdonságait atomi felépítésük törvényszerűségeire, a részecskéik között fellépő kölcsönhatásokra. Közben azonban számtalan olyan jelenség gyakorlati alkalmazására és empírikus kutatására is sor került, amelyeknek elméleti alapjait előzőleg csak nagy vonalakban sikerült a tudománynak tisztáznia.

Lényegében ma is ugyanez a helyzet, legfeljebb ismereteink határai tolódtak kijjebb. A Mikroelektronika és elektronikai technológia c. könyv egyik szerzője ma is teljes joggal írhatja, hogy a félvezetők technológiai kérdéseinek vizsgálatában azért van olyan nagy szerepe a technológiai folyamatok szimulálásának, mert „még ma sem ismert valamennyi lépés egzakt fizikai-kémiai háttere". A folyamatok szimulálása biztosíthatja a lehetőséget a fizikai-kémiai modellek tökéletesítéséhez és a meglévő eszköztár folyamatos tökéletesítéséhez — írja. Gyakran nemcsak az értékük, hanem még mibenlétük is bizonytalan azoknak a bemeneti paramétereknek, amelyek megadása nélkül lehetetlen volna megoldani a gyakorlatban felmerülő problémákat. Tudjuk, hogy fontos, hozzávetőleges pontossággal még az értékét is sejtjük, csak azt nem tudjuk, hogy micsoda...

A harmincas-negyvenes évek technikai csodái között csak Hamupipőke volt a kristálydetektoros rádiók egyenirányítójaként használt galenitkristály, a kristályos alakú ólomszulfid. Ki hitte volna akkor, hogy nem a büszke elektroncsövek hexódái, heptódái, pentagrid keverői, októdái, enneódái fejlődnek tovább, hanem primadonnává válik a negyvenes évek végére Hamupipőke unokahúga, a tranzisztor?

Az ötvenes években kezdtek komolyabban érdeklődni a szervetlen kémia kihasználatlan lehetőségei iránt. Az első tranzisztorok egy természetes anyagot, a germániumot használták a galenitkristály helyett, ebből azonban nem lehetett nagyobb mennyiséget kitermelni, s nagy hibája volt, hogy gyorsan elromlott, mert a felülete nem stabilizálható.

A sziliciumvegyületeket akkoriban még a természet selejtjének tartották. Majdnem ugyanannyi változatuk volt, mint a szénvegyületeknek, de csupa jellegtelen, arctalan képződmény. Mivel azonban a periódusos rendszerben a szilicium éppen a germánium és a szén közé esett, egyetemi laboratóriumokban elkezdték vizsgálni anyagi tulajdonságait. Kiderült, hogy nem egyetlen anyagnak, a germániumnak a különleges adottsága, miszerint tranzisztort lehet belőle készíteni, hanem például a sziliciumnak is, de ezenkívül is számtalan olyan félvezető anyag és vegyület van, amely hasonlóképpen viselkedik.

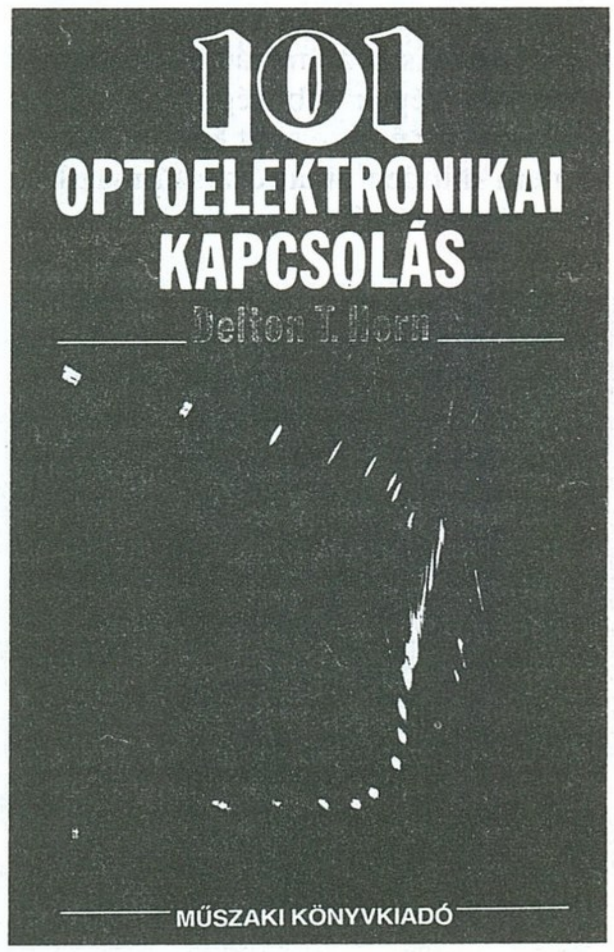
Így lett fokról fokra a szilárdtest-fizikából, az egyik „legtisztább" elméleti diszciplínából korunk legfontosabb gyakorlati tudományága.

D. T. Horn:

101 optoelektronikai kapcsolás

Műszaki Könyvkiadó, 1992
192 oldal, 790 Ft.

A könyv első része négy rövid fejezetben ismerteti az optoelektronika alkatrészeit: a fényelemeket és fotoellenállásokat, a fototranzisztorokat és rokonságukat, a különböző fénykibocsátó eszközöket, valamint a száloptikákat és optocsatolókat. A könyv nagyobbik része azonban különböző kapcsolások leírásával foglalkozik a legkülönbözőbb alkalmazási területekről. A napelemmel működő akkumulátortöltőtől kezdve a túlfeszültségjelzőkön és a fénnyel vezérelt jelfogókon, motoro-



kon, világításon, hanggenerátoron keresztül a LED-villogókig és a különböző ellenőrző és mérőberendezésekig számtalan érdekes és ötletes kapcsolást ismertet a könyv a hozzá szükséges alkatrészek pontos felsorolásával. Játékos kedvű olvasóinknak külön felhívom a figyelmét, hogy külön fejezet foglalkozik az elektronikus játékokkal.

Az optoelektronika ma a mikroelektronika egyik igen perspektivikus területének tűnik. Aki ezeket a kapcsolásokat kipróbálgatja, játékos formában juthat el arra a szintre, ahol már a komoly alkotómunka kezdődik. Érdemes elmenni a szomszédba az ötletekért!

V. Nagy Edit

Ismeretlen ismerős: Siemens Nixdorf

A környezetkímélő számítógép(gyár)

Új rovatunk, a VIZIT olyan szakmai beszámolóknak ad ezentúl keretet, amelyek érdekes információkat, újdonságokat, mélyebb elemzéseket kínálnak valamelyik cégnél tett látogatásunk alapján. Az alábbi cikk a Siemens Nixdorf augsburgi és paderborni gyárában, a 62. oldalon lévő írás pedig az IBM milánói laboratóriumában látottakból igyekszik ízelítőt adni.

Magyarországon közel ugyanannyi „neves" mint „névtelen" számítógép kerül forgalomba: az összeszerelt 'no name' alkotások zöme távol-keleti, míg a márkás gépek többsége valamelyik amerikai cég égisze alatt készül — még ha nem minden alkatrész származik is az óceánon túlról.

Meglepő sorrend

A márkanevek ismertsége, a kiállításokon, szaküzletekben vagy munkahelyeken látottak alapján „saccolva" itthon kevesen akadnának, akik az európai értékesítés élcsoportjának felsorolásából ne felejtenék ki a Siemens Nixdorfot. A Siemens név Magyarországon elég jól cseng, háztartási gépei, orvosi műszerei, sokféle ipari berendezései és mainframe számítógépei a jó hírnevet évtizedek alatt megalapozták, vele kapcsolatban a PC-re mégis nagyon kevesen gondolnak. Pedig noteszgépektől a vállalati szerverekig a személyi számítógépek teljes skáláját gyártják. Ami pedig a rangsort illeti, 1995. III. negyedévében a „top 10" tagjai a következő arányban részesedtek az európai PC-eladások árbevételből, amely összesen 7,6 milliárd USA dollárt tett ki:

1. Compaq	15,5%
2. IBM	11,1%
3. Apple	6,3%
4. Siemens Nixdorf	5,4%
5. Hewlett-Packard	5,3%
6. Toshiba	4,0%
7. Dell	3,9%
8. Escom	3,7%
9. Highscreen	3,0%
10. Digital	2,9%
A többi együtt	38,9%

Ugyanebben az időszakban az Európában eladott PC-k darabszámának (összesen 3,08 millió gépnek) a megoszlásában a Siemens Nixdorf már „csak" az 5., mert a viszonylag alacsonyabb átlagárú Escom öt hellyel előbbre rukkol. (Az élcsoport második felében is több ponton eltér a darabszám-sorrend, szintén az olcsóbb gépeket gyártók javára.) A jelenlegi trendek alapján mégis reálisnak tűnik, hogy a Siemens Nixdorf 1996-ban a PC-k értéke és darabszáma szerint egyaránt elérje a megcélzott harmadik helyet. (Az első két helyen nagy előnnyel vezető Compaq és IBM megszorítására csak távlatban van esélye.) Az 1995. szeptember 30-án lezárult gazdasági évben a Siemens Nixdorf 657 800 PC-t gyártott, a következő gazdasági évre pedig 1 millió darab PC előállítása a kitűzött cél. A Siemens Nixdorf márkájú PC-k száma Magyarországon 1995-ben az összes eladott PC-nek csak mintegy 2 százaléka, kb. 2500 db volt, de az előző évinek ez már a négyszerese.

Magyar monitorszállítás

A tapasztalatok azt mutatják, hogy a Siemens Nixdorf minőségileg egyenrangú az amerikai markákkal, a gépek átlagára pedig a hasonló tudású konkurenseképez képest 2-3%-kal alacsonyabb. A német alapossággal kialakított minőségellenőrzésen túlmenően olyan „apróság" is jelzi az igényességet, hogy az Inteltől vásárolt (ma már többnyire Pentium) processzorokat mindeddig közvetlenül az Egyesült Államokból importálták, mert az írországi gyár korábban nem érte el a Siemens által megkövetelt 99,7%-os megbízhatósági szintet. Csak a közelmúltban született egyezség arról, hogy a megfelelőnek talált minőségvizsgálati eredmények és az Intel garanciája alapján ezentúl európai forrásból szerzik be a processzorokat, s így a Siemens Nixdorf gépei megkaphatják az európai származási bizonyítványt, valamint a kedvezményes vámkezelést biztosító „Eur-1" bizonylatot.

Jelentős „európásító" lépésre szánta el magát a Siemens Nixdorf egy másik fontos tartozék beszerzésében is. Lapzártakor már aláírás előtt állt az a szerződés, amelynek értelmében az eddig Délkelet-Ázsiából importált monitorok jelentős hányadát a Philips magyarországi gyárai fogják szállítani. Ha az üzlet beválik, ha a magyar termék megfelel a követelményeknek, évente több százezer (egyelőre 14 és 15 collos) monitort exportálhatunk az SNI számítógépeihez.

A Siemens Nixdorf gyári bemutatótermében elhelyezett termékek közül feltűnt a multimédiára alkalmas gépek kialakításának sokféle variálása, különösen a legújabb, Scenic elnevezésű sorozatban. Kívülről kevésbé látható a gépek racionálisabb belső konstrukciója, ami elsősorban a bővíthetőséget, a modularitást, a kevesebb alkatrészből történő szerelhetőséget szolgálja. Ha az 1987-es Siemens PCD-2 típusú gépet összehasonlítjuk az 1995-ös PCD-4L géppel, elég szembetűnő a változás:

	PCD-2	PCD-4L
Súly	16 kg	6 kg
Szerelendő alkatrész	87 db	14 db
Szerelési idő	33 perc	8 perc
Szétbontási idő	18 perc	5 perc
Vezetékek száma	13 db	5 db
Billentyűzetalkatrész	155 db	6 db

Közismert, hogy a technika világában kevés kivétellel általában az a költségesebb megoldás, amely a környezetet jobban kíméli, és az bizony élesen ütközik a (rövid távú) profitérdekkel. A messzebbre előrelátó cégek mégis a nehezebbik utat választják. Így tett a Siemens Nixdorf is. A számítógép működése közbeni környezetkímélés volt az egyik cél. Például energiatakarékos PC-jük bekapcsolt, de éppen nem használt állapotban automatikusan átkapcsolja magát várakozó üzemmódra (30 watt alatti fogyasztásra). Augsburgban külön tesztlabort építettek fel minden

új géptípus elektromágneses sugárzásának ellenőrzésére is (annak méteres érzékelő tüskékkel „kibélelt" elektronikus süketszobája látható a mellékelt fényképen). A laborban mérik, pontosan „feltérképezik" a gép és a monitor által kibocsátott elektromágneses sugárzásokat (output), és azt is, hogy mennyire érzékeny a számítógép a külső elektromágneses zavarokra (input).

A kék angyal visszatér

Számítógépeiben a Siemens Nixdorf az elsők között csökkentette minimálisra az újrahasznosíthatatlan alkatrészeket és anyagokat, illetve azokat, amelyek ismételt feldolgozásakor káros melléktermék (pl. mérgező gáz) keletkezik. A kopásállóságon kívül ezt a célt is szolgálja például, hogy a billentyűzet betűit festés vagy matricák alkalmazása helyett lézersugárral égetik be magába az anyagba.

Legérdekesebb és szinte egyedülálló megoldás, hogy a paderborni újrafelhasználási központ egész Nyugat-Európából visszaveszi és szakszerűen szétszedi a kiöregedett Siemens Nixdorf számítógépeket és nyomtatókat. 1994-ben 4497 tonna ilyen begyűjtött „roncsnak" 82 százaléka volt újra felhasználható, az 1995. évi 5383 tonnából 69%-ot tettek ki a fémek és más feldolgozható anyagok, 17%-ot az újra felhasználható alkatrészek, és már csak 14% volt a hulladék. Az újrahasznosítási arányt szeretnék a következő évben 90 százalék fölé emelni.

Annak ellenére, hogy az elhasznált, kiöregedett eszközöknek a termelésbe való visszaforgatása Magyarországon még nem szerveződött meg, tehát nincs igazán összehasonlítási alapunk, egy kicsit talán a jövő felé kacsintva érdemes egy pillantást vetni a németországi visszavásárlási árakra:

Kisebb készülék 10 kg-ig (tartozékok nélkül)	20,– DM
Nagyobb készülék 40 kg-ig (tartozékok nélkül)	30,– DM
Monitor és billentyűzet	35,– DM
Nyomtató munkaállomás 50 kg-ig	62,– DM
Rendszerek 200 kg-ig	165,– DM
Mainframe, mágnesmemória 1200 kg felett	970,– DM
Adathordozó	1,20 DM/kg
Kábel	0,35 DM/kg

A szétszerelési szempontokra ma már az új termékek megtervezésekor eleve odafigyelnek. A Bayer AG-val együttműködve olyan technológiát fejlesztettek ki, hogy a régi műanyag ház megőrölve 25 százalékban hozzákeverhető az új készülékházak gyártási alapanyagához.

Nemcsak a terméknek, hanem magának a gyártási folyamatnak környezetkímélő kialakítására is találtak a Siemens Nixdorfnál követésre érdemes megoldásokat:

— A termeléshez csak visszaforgatott, állandóan újra tisztított vizet használnak, ipari célra tehát nem vesznek fel a hálózatból friss vizet.

— Festés helyett oldószert nem igénylő eljárással, elektrosztatikus porszórással készítik a bevonatokat.

— Hulladékmentes technológiák bevezetésével szinte semmit nem hagynak veszendőbe menni.

A gyártás és a produktum környezetkímélő tulajdonságai révén a PC-k közül elsőként a Siemens Nixdorf személyi számítógépei nyerték el ez itt látható „kék angyal" emblémát, a környezetvédelem minősítő jelzését.

Az előbbiekben felsoroltakon kívül ehhez olyan kritériumoknak is eleget tettek, mint a műanyag részek jelzésekkel való ellátása, alacsony zajszint, kadmium és higany nélküli elemek használata, kézikönyvek klórmentes papírra való nyomtatása stb.

Nem csoda tehát, hogy 1995-ben Németországban a számítógépek környezetkímélő kialakítását mutató hivatalos pontrendszer skáláján a Siemens Nixdorf érte el a legjobb minősítést, 105 ponttal, utána következett a Hewlett-Packard (79), a NEC (78), az AT&T (76), az Acer (60), az IBM (59), a Compaq (52), a Vobis (24), az ASI (18), a Peacock (12), míg az átlagot jelentő nulla pont alatt helyezkedett el az Olivetti (–10), az Apple (–32) és a ComTech (–88).

Faklen Pál



Az IBM és a kliens/szerver architektúra

Nyitottan — olcsón

A kliens/szerver rendszer nem megváltó, nem is csodaszer, hanem csupán egyike azoknak a megoldási módoknak, amelyeket a feladatok hívtak életre. Ráadásul nem is a legolcsóbbak közül való. Ilyen gondolatokat is kelthetett az a szakmai találkozó, amelyre az IBM milánói regionális központjában került sor a közelmúltban.

A milánói előadások és bemutatók célja az volt, hogy a kliens/szerver fogalomkörrel kapcsolatos IBM-álláspontot árnyaltabban ismertessék meg, kezdve a globális, piaci megközelítéstől, a munkacsoportok és a Lotus Notes problematikáján és a rendszermenedzsmenten keresztül egy konkrét banki projektig.

Európában — elsősorban az üzleti alkalmazások körében — évről évre jelentősen nő azoknak a cégeknek a száma, amelyek a kliens/szerver architektúra bevezetése mellett döntenek. Jelenleg azonban még 50%-ra sem tehető a kliens/szerver megoldások aránya az össz-európai piacon. A feladat, a kihívás tehát a szállítók számára adott, ugyanakkor nem szabad figyelmen kívül hagyniuk azt a tényt, hogy a meglévő rendszerek átszervezése tetemes költségeket ró a felhasználókra.

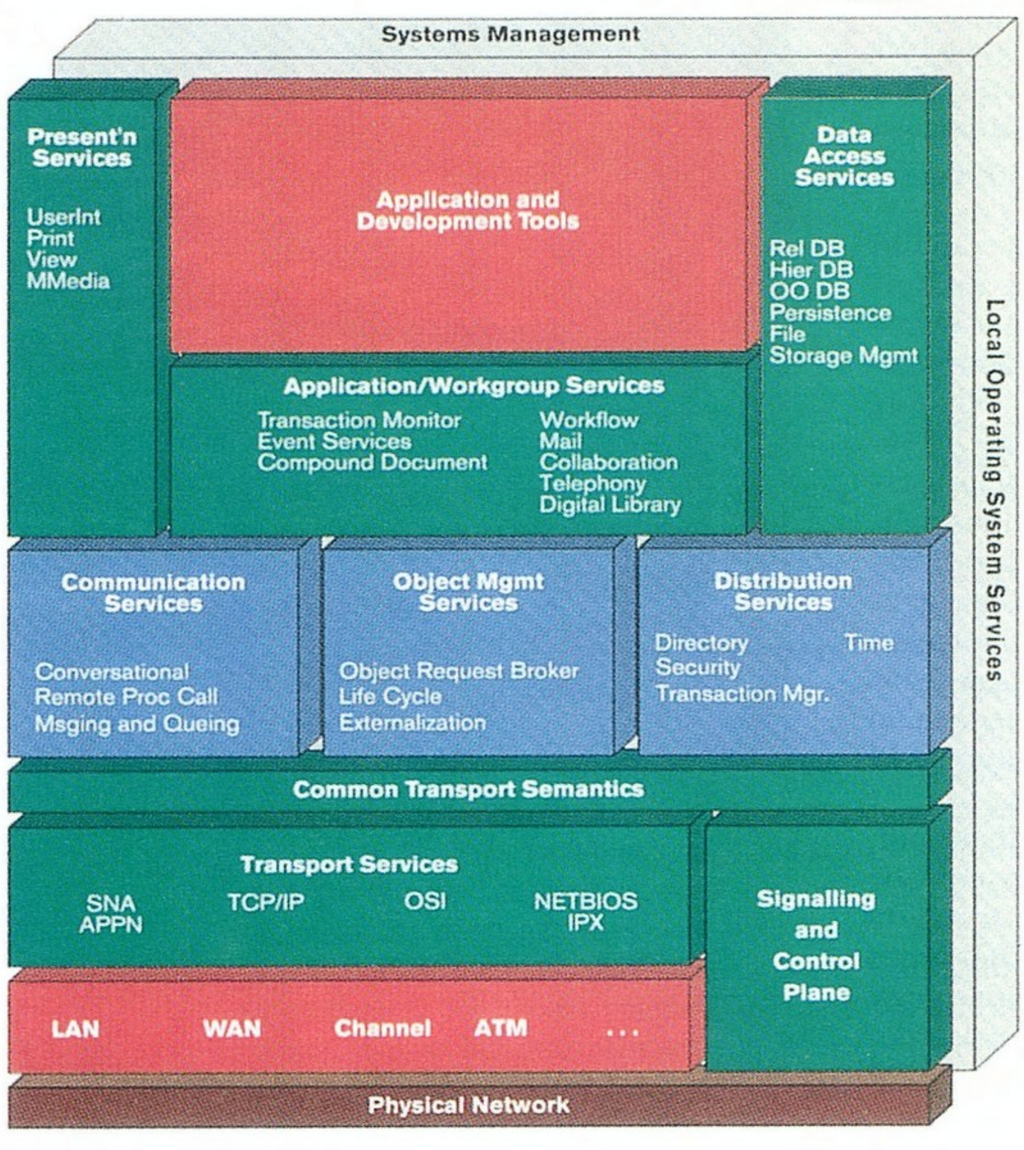
Az IBM álláspontjának fő ismérve — a költségtakarékos megoldás lehetőségét keresve — a nyitottság, a heterogén rendszerek egymás mellett létezésének, együttműködésének a biztosítása. Nem elegendő egy kliens/szerver rendszert kínálni, mondják az IBM-nél, mindezt úgy kell megvalósítani, hogy az a felhasználóknál a legkisebb ráfordítást és a legjobb megtérülést eredményezze. A felhasználók egyszerűséget, nagy teljesítményt és az alkalmazások széles skáláját várják el a kliens/szerver rendszerek szállítóitól. (Egyesek szerint az így megfogalmazódó feladat annyira komplex és olyan nehezen teljesíthető, mintha kifejezetten az IBM-nek találták volna ki.)

Az IBM még 1991-ben szervezte meg és rendezte be Milánóban azt a kísérleti célú laboratóriumot (Multivendor Laboratory), amely a különböző gyártók rendszereinek vizsgálatát szolgálja — főként azok együttes működtetése szempontjából. E laboratórium a világ számos helyén kiépített Open System központokkal együtt összegyűjti a különböző gyártóknál különböző feladatok megoldására készült (hálózati) rendszerek működési tapasztalatait, s ebből az adatbázisszerű forrásból félig vagy teljesen kész megoldás is választható egy-egy újonnan megfogalmazódó számítástechnikai feladatra.

Az IBM kliens/szerver célokat szolgáló termék- és szolgáltatáspalettájának két kulcseleme van: a SystemView mint rendszermenedzsment szoftver és az Open Blueprint mint rendszerszervezési technológia. Mindkettőnek az a funkciója, hogy a maga területén — a felhasználó számára láthatatlanul — rendet teremtsen a kisebb-nagyobb hálózatok dzsungelében. A SystemView-nak elkészült az OS/2 platformot támogató változata, amellyel nemcsak a PC-s környezet jobb menedzselhetősége valósítható meg, hanem a szabványos SNMP protokollon keresztül más SystemView rendszerekkel is (AIX, OS/400, MVS) szilárd kapcsolat építhető ki a vállalati szintű integráció érdekében.

A SystemView legfontosabb szolgáltatásai közé a valós idejű monitorozás (a hálózati menedzser előbb értesül a hibáról, és előbb oldja meg, mint ahogy a felhasználó észrevenné), a szoftvermegosztás, az automatikus leltározás (a felhasználó által installált alkalmazások követésére) és a távoli vezérelhetőség tartozik. Egy elemző szerint a SystemView for OS/2 alkalmazása révén a szoftverinstallációs költségek 50%-kal, a támogatás költségei legkevesebb 33%-kal csökkenthetők.

Az Open Blueprint koncepció egy sor olyan szolgáltatást ír le, amelyek szükségesek az alkalmazások eléréséhez egy nyitott, megosztott, heterogén környezetben. Ezeknek a szolgáltatásoknak nem kell minden hálózati platformon létezniük, de a hálózat minden pontjáról elérhetőknek kell lenniük. Az Open Blueprint felépítését a mellékelt ábra mutatja.



Az előadások sorát egy bankfiók-reautomatizálási projekt eredményeinek ismertetése zárta. A Scala 6 projekt keretében arra törekszenek, hogy egy bankfiók teljes számítástechnikai rendszerét egyetlen nap alatt lehessen átalakítani (ez már működő technológia!), s hogy mindez lényegében „sorozatgyártássá" alakuljon, vagyis minden napra jusson egy-egy bankfiók-rekonstrukció, jelenleg ugyanis a heti 1-2 fiókos tempó a gyakorlat.

Varga János

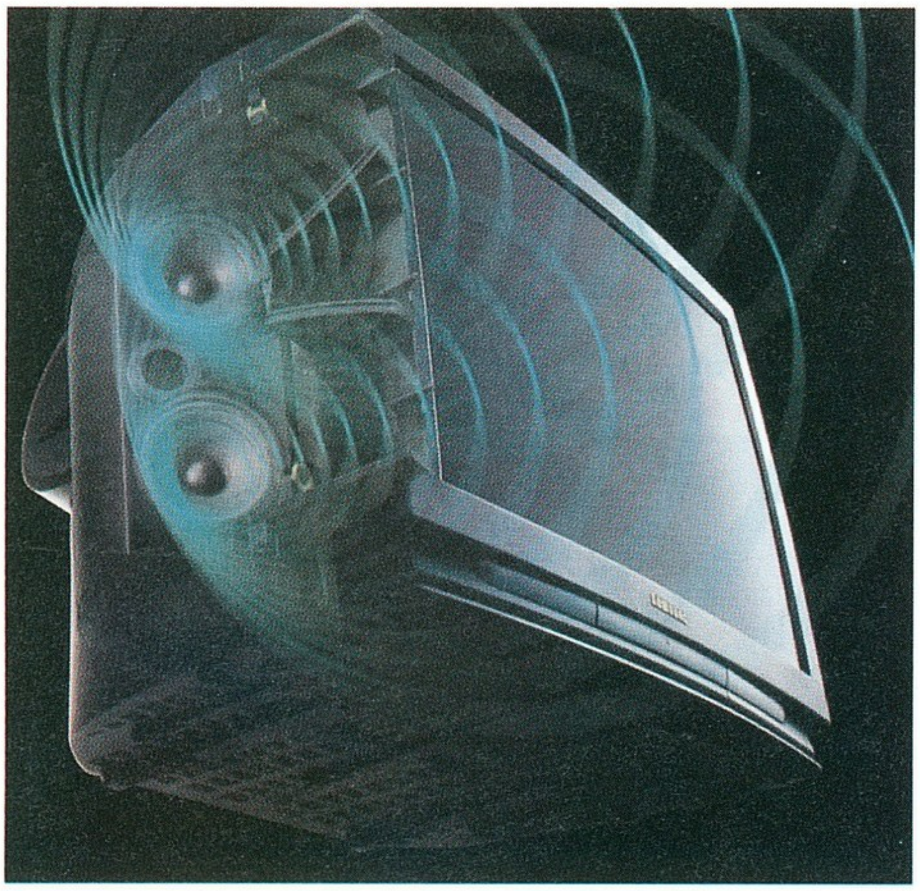
Nagyképűség?



A Samsung nemes egyszerűséggel a világ legjobbjaként, a „World Best" jelszóval vezeti be új termékét, az Ultra Bio Vision tévét. A jelző elsőre igencsak nagyképűen hangzik, bár úgy tűnik — és ez nem elhanyagolható szempont —, van is mire büszkének lenniük. A Samsung Ultra Biójának leg-leg jellemzője a görbületet kifejező 2.5 R. Ez az eddigi csúcstartó „szuper lapos" képcsövek 2.0-s uralmának vet véget (ugyanis minél nagyobb a szám, annál közelebb van a síkhoz a képcső képfelülete). Ez éppen a nagykép(ernyőj)ű készülékeknél fontos: a képernyő sarkaiban is minden korábbi megoldásénál tisztább, élesebb képet eredményez.

Egy tévékészülék bemutatása általában nem túl érdekes egy számítástechnikai lapban, de ez a bemutatkozás megítélésünk szerint túlmutat saját kategóriáján. Az Ultra Bio ugyanis jelentős állomás lehet abban a fejlesztésben, amelyet leginkább a Mitsubishi, a Philips, a Grundig, sőt 500 dolláros gépének beharangozásával talán az Oracle munkája is fémjelez, mert a több funkciót integráló, intelligens kijelzők már a jövő legfőbb (számítás)technikai eszközének előfutárai.

Az Ultra Bio „repertoárjában" kiemelésre méltó a színveszteség korrigálására szolgáló automatikus vezérlőáramkör, a gyenge vételi viszonyok között is jó minőséget garantáló alacsony zajszintű erősítő, valamint a hi-fi minőségű hang — 60 wattos terhelhetőséggel. A korszerű készülékekkel szemben szinte alapkövetelményként támasztott igénynek, a környezetbarát jellegnek is teljes mértékben



eleget tesz a Samsung új „tigrise": a névadó bio funkció pedig úgy valósul meg, hogy a készülék a káros elektromos sugarak csökkentése mellett az élő szervezetek számára hasznos infravörös sugarakat is kibocsát.

Ugyanazt másképp!

Patinás holland cég helyezte új alapokra magyarországi tevékenységét saját vállalat létesítésével. (Igaz, „történelmileg" egy kicsit rosszul időzítve, éppen az Ajax–Ferencváros „háborúskodás" idején.) A 120 éves Océ évtizedek óta ott van a magyar piac szereplői között, eddig disztribútora, a Reprocom-Datacom kínálta szolgáltatásait, mostantól az Océ Hungária Kft veszi át a feladatot.

Az Océ joggal állíthatja magáról, hogy azon ritka reprográfiai cégek közé tartozik, amelyek mindhárom szakterületen — másolás, nyomtatás, plottolás — ott vannak a szolgáltatók élmezőnyében.

A gyártó-forgalmazó egész termékskálájának jelentős piacbővítését reméli Magyarországon, de a legtöbb reményt a 3045-ös jelzésű másolóhoz, az analóg és digitális másológépek új generációjának első termékéhez fűzi. Ez a család, amely a korábbi 1900-ast váltja fel, az ígéretek szerint ötvözi az előd könnyű kezelhetőségét („csak lépj oda, és másolj") a széles felhasználási lehetőséggel és az igen barátságos árral. Néhány jellemző (szolgáltatás): egyszerű kezelőpanel, levegőszétválasztásos automata adagolórendszer (percenként 45!), négyrekeszes papírtároló (összesen 3000 ív, „sorba kötött" adagolórekeszek révén folyamatos papírellátás), kétoldalas másolás egy lépésben, margóállítás, könyvmásolás (!), függelékkészítés.

A 3045-ös környezetvédelmi megfelelésére egyetlen érv: kiérdemelte a szigorúságáról nevezetes német környezetvédelem Kék Angyal védjegyét.

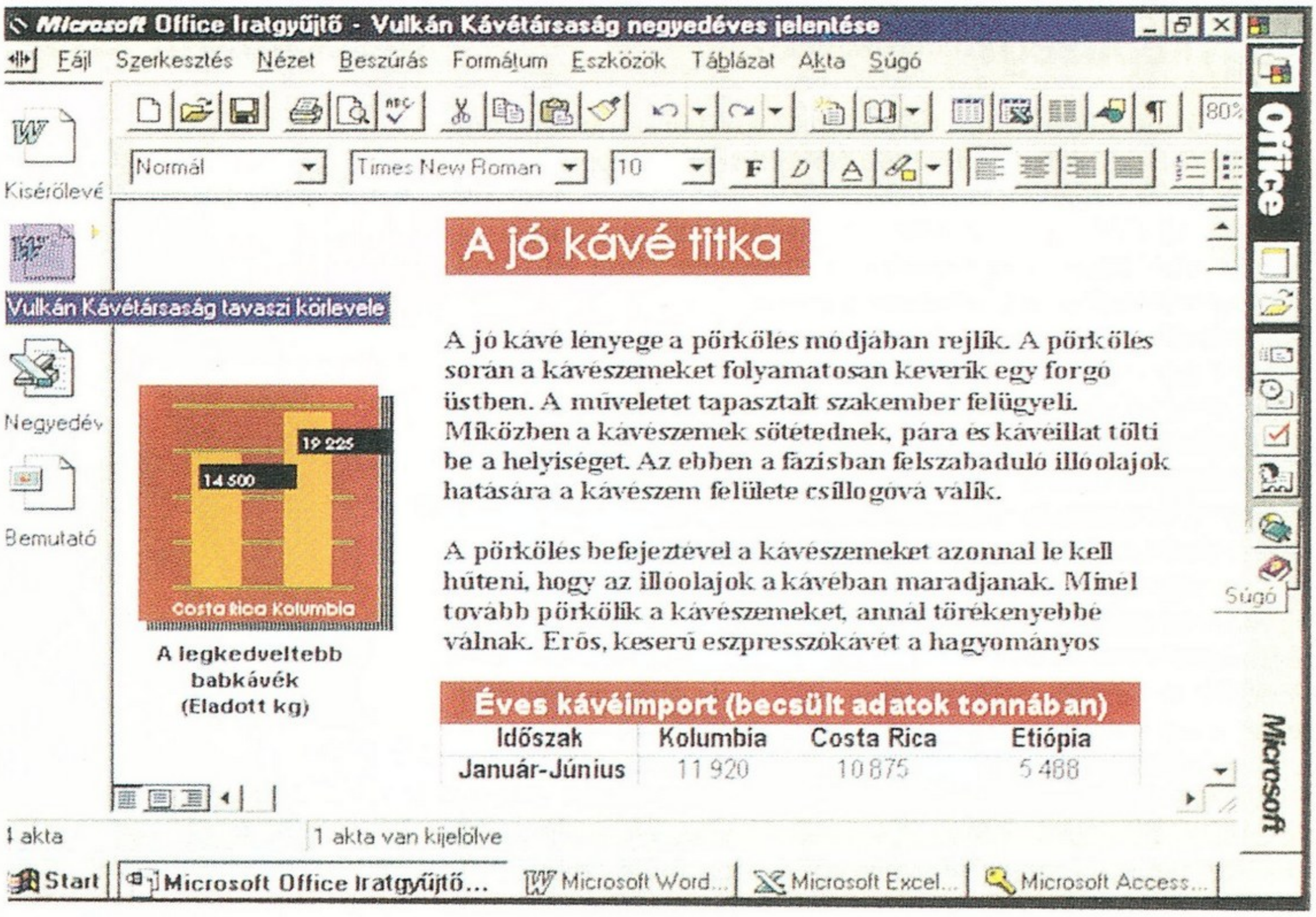
Akciónap II/95

Mikulás után két nappal volt a Microsoft második magyar nagypénteke. Ezúttal az Office magyar változatának premierjére került sor, s az igencsak kedvező bevezető ár (lénye-

gében a leendő végleges ár felét kellett leszurkolni) méltán eredményezett nagy forgalmat: például az egyik budapesti viszonteladónál záróráig 75 darab talált gazdára.

A reklámanyagok a 32 bites lehetőségek kihasználásának eredményeként egyes alkotóelemek működésében jelentős teljesítménynövekedést ígérnek — a Word, az Excel, a PowerPoint és a Schedule+ a professzionális változatban kiegészül az Access-szel, a profi CD-s verzióban pedig a Bookshelffel is.

Örvendetes, hogy az új Office 95 — szemben a korábbi változat *lényegében* magyar megjelenésével — már *teljesen* magyar nyelvű. Egységes kezelői felületen keresztül használható valamennyi program, így a „házon" — azaz az Office-on — belüli alkalmazások adatkommunikációja is lényegesen leegyszerűsödik.



S bár a végfelhasználók — és a viszonteladók — csak meglehetősen macerás úton juthattak a termék végleges változatához, mert az akciónapon például csak korlátozott licencű CD-t kaptak (szigorú figyelmeztetésekkel körítve), és azt mintegy bónként válthatták be a a később elkészült végleges termékre —, de legalább sikerült elkerülni a Win95Hun-szindrómát, azaz ezúttal *mindenki* hozzájuthatott a termékhez az akciónapon. A gyártókapacitás szűkös volta miatt ugyan kellett egy kicsit várniuk...

Ultra

A Sun közismert SPARC rendszereit világszerte elsősorban az üzleti vállalkozások, oktatási intézmények, kormányzati szervek használják technikai, kereskedelmi, ipari és szoftverfejlesztési alkalmazásokhoz. Hasonló kört céloznak meg annak az új, hálózatközpontú architektúrának az első képviselői — a munkacsoport-szerverek —, amelyeket az UltraComputing elnevezés fűz egybe. A Sun kijelentése szerint az UltraComputing elérhető áron biztosít szuper-computing képességeket. Kitűnő grafikai teljesítményű, illetve nagy sávszélességű hálózati szolgáltatást nyújt „egyszerű" desktop rendszerekben is.

Az UltraComputing — kategóriájában elsőként — a „crossbar switch" technológiát alkalmazza: annak érdekében, hogy rendkívül kis késedelemmel lehessen sokféle adatutat használni, összekapcsolódik a CPU, a memória, az I/O, valamint a rendszer hálózata. Az UPA-nak (UltraPort Architecture) nevezett kapcsolási technológia a hagyományos memóriabuszt váltja fel, alkalmazása révén biztosítható a 64 bites UltraSPARC processzor teljesítményének kihasználása. Az UPA révén a fő processzorlapra integrálódik a 64 bites feldolgozás, a 64 bites I/O, a gyorsdiszkes I/O, valamint a 100 Mbit/s-os Ethernet. A számos „leg" közül — ezek között akadnak a konkurens termékek jellemzőihez viszonyítva szinte hihetetlen deklarált adatok is — mindenképpen kiemelésre méltó e havi vezértémánk vonatkozásában a Mitsubishivel közösen kifejlesztett 3D-RAM technológia, amely kulcseleme az új architektúrának.

Az elsők között bejelentett szerverek 19 000 és 29 500 dollár közötti áron kerülnek majd forgalomba, 143–167 MHz-es UltraSparc processzorral, 32–64 Mbájt RAM-mal, 2–től kétszer 2,1 GB-osig választható harddiszkkel. Operációs rendszerük a Solaris 2.5.

A Sun új irányvonalához számos szoftvercég (például: Adobe, Ansys, Informix, Netscape, Oracle, Parametric, SAS, Sybase) csatlakozott már, kijelentvén: alkalmazásaik legkésőbb 1996 első negyedévének végére rendelkezésre állnak ezen a platformon is.



Újévi meglepetés hihetetlen árakon

csak nálunk, csak viszonteladóknak



ETHEREXPRESS PRO HÁLÓZATI KÁRTYACSALÁD

- 100/10 Mbps hálózatokhoz egyaránt
- Combo és TP kivitel
- gyors installáció
- Token család is, 4/16 Mbps

NETPORT EXPRESS PRINT SERVER

- 3 printer csatlakoztatható egyidejűleg a hálózathoz
- gyors installálás
- multi-protocol (NetWare, Unix, Lan Manager, LAN Server, Apple Talk)

intel®

COMPUTER 2000 MAGYARORSZÁG

1133 Budapest, Váci út 110. Tel.: 267-1888, Fax: 267-1901

INFORMÁCIÓKÉRÉS: 0105 ▲

Computer PANORÁMA aktuális

Windows® 95

Egy könyv,

- amelyből megtudhatja: miért jó, kezes és barátságos az új operációs rendszer...
- amely több mint egy használati utasítás...
- amely bevezet a 95-ös verzió új lehetőségeinek, fogalmainak, technológiájának sokszínű világába...
- amely megkönnyíti az átnyergelést a Windows 3.1-ről...

A tartalomból:

• Kérdések és válaszok • A megváltozott kezelői felület
• MS DOS-programok • Nyomtatás • Fájl- és diszkrendszer • Kommunikáció • Plug and Play
• Multimédia • Az operációs rendszer újdonságai

Megrendelhető levélben, faxon vagy telefonon a Kiadónál.

Cím: 1077 Budapest, Wesselényi u. 17. IV. em. • Telefon: 322-4248, 122-9556 • Fax: 322-1032

INFORMÁCIÓKÉRÉS: 0106 ▲

10 éves az Info-Katalógus!!!



A katalógus tizedik születésnapja Partnereink érdeme is. Köszönjük!

A katalógus új formája a ZSEBKÖNYV.

Válasz-levelezőlapjainkkal elősegítjük a közvetlen és gyors kapcsolatfelvételt az eladó és a vásárló között.

Partnerkereső fejezetünk célja a kereskedelmi és az üzleti kapcsolatok bővítése.

[gyertya]-edik éve,
[gyertya] ezer példányban,
most a [gyertya]-edik újdonsággal: az ügyfélkártyával!

([gyertya] = tíz)

Előfizetünk az **Info-Katalógus '96 I.** félévi számára **1200 Ft-os áfás áron**, amelynek összegét a mai napon átutaltuk a MADE-INFO KFT. **Postabank 11991102/021-16565** számú számlájára! (A Kiadó a pénz beérkezése után azonnal számlát küld, és a kiadványt megjelenés után postázza.)

Cégnév: .... **Ügyintéző:** ....

Pontos cím: ....

Telefon: .... **Telefax:** ....

UA